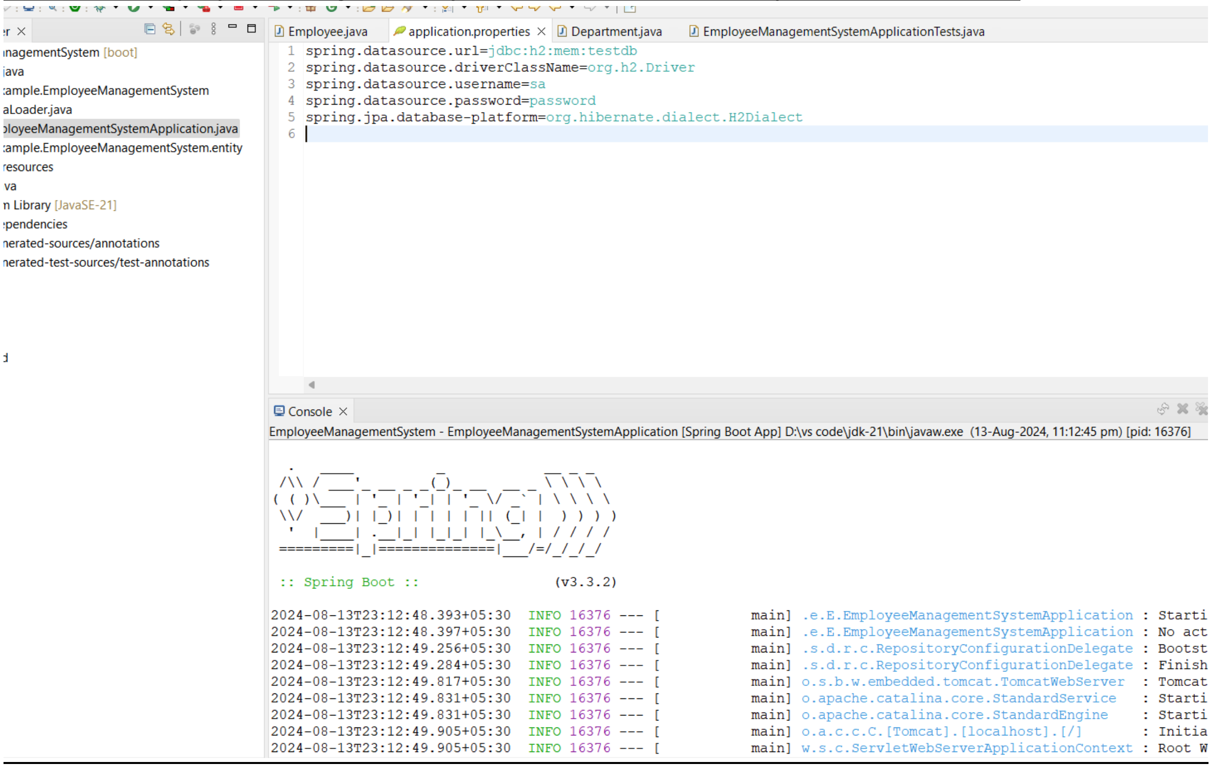
# Digital Nurture 3.0 I Deep Skilling (WEEK 3 SOLUTIONS)

**<u>Superset ID:5021661</u>**

**<u>Name: Priya Hazra</u>**

## Exercise 1: Employee Management System - Overview and Setup Business Scenario:

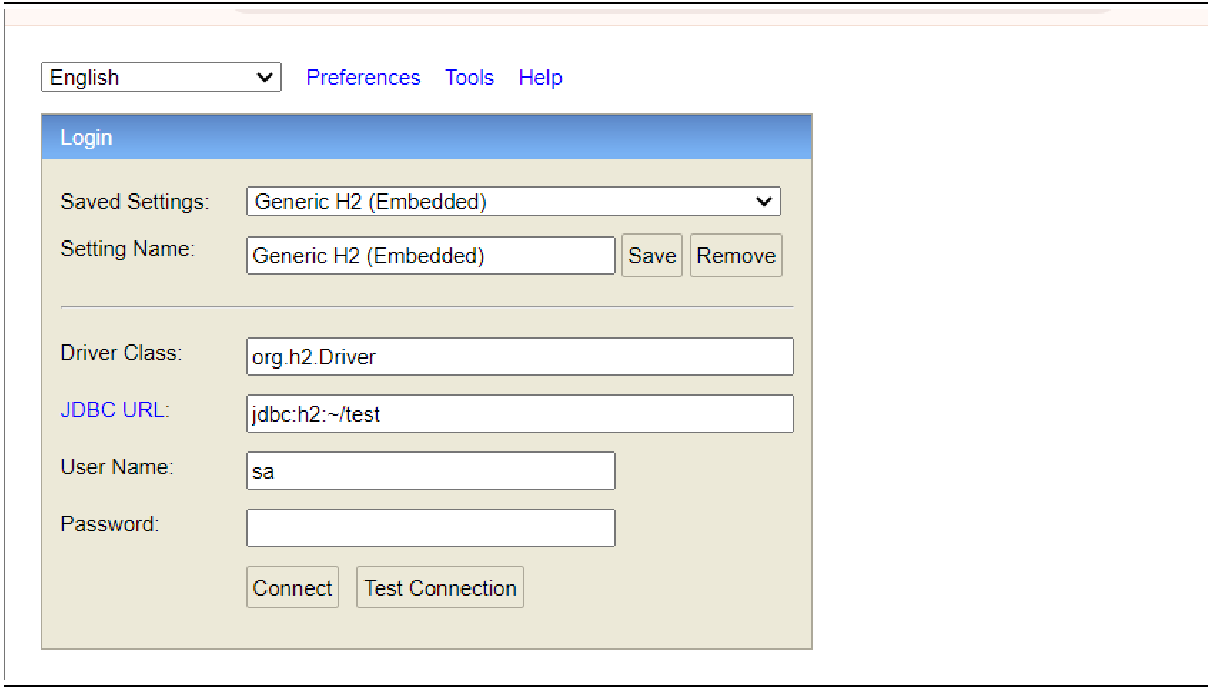


nagementSystem [boot]
java
ample.EmployeeManagementSystem
aLoader.java
ployeeManagementSystemApplication.java
ample.EmployeeManagementSystem.entity
resources
va
n Library [JavaSE-21]
pendencies
nerated-sources/annotations
nerated-test-sources/test-annotations
Employee.java
application.properties
Department.java
EmployeeManagementSystemApplicationTests.java
Console

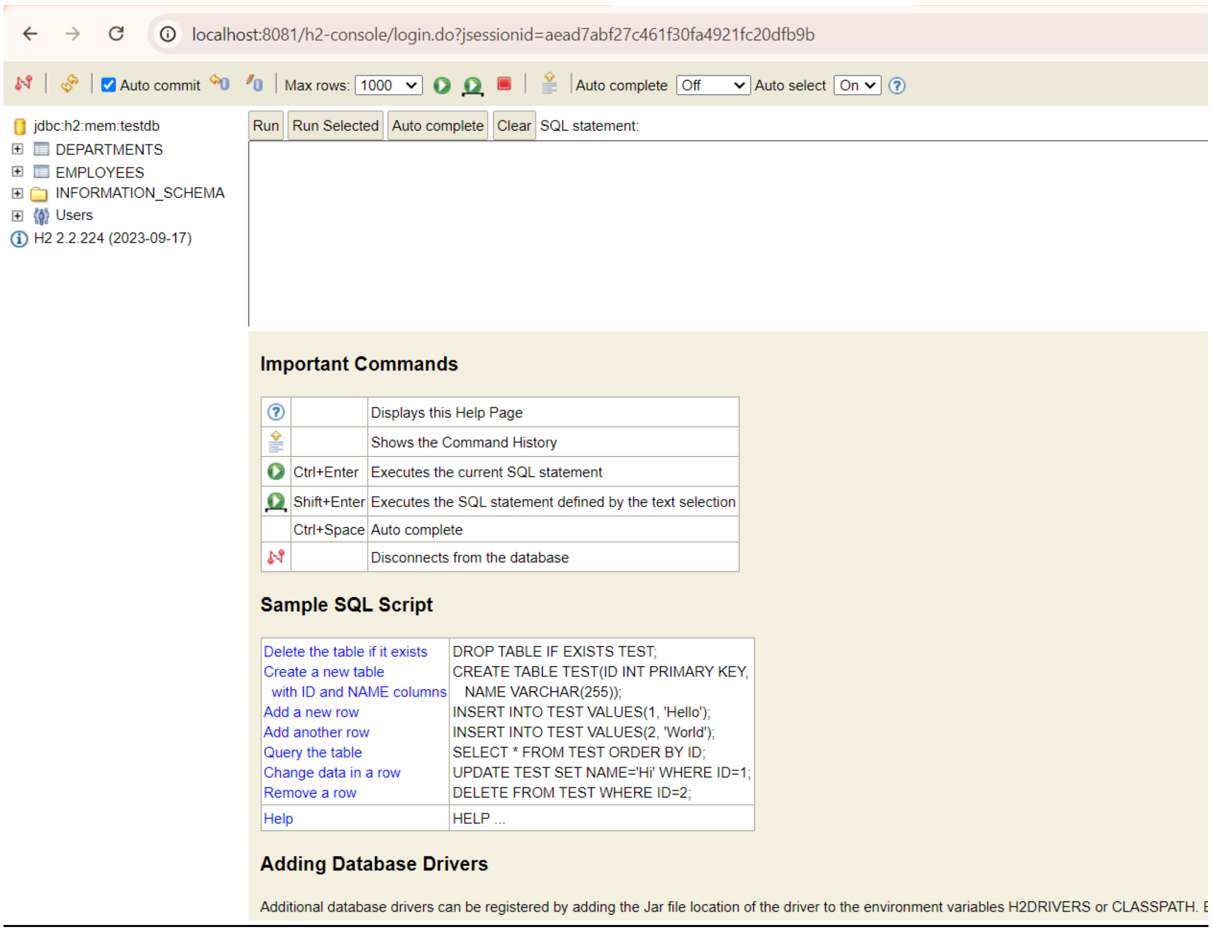
```
spring.datasource.url=jdbc:h2:mem:testdb
spring.datasource.driverClassName=org.h2.Driver
spring.datasource.username=sa
spring.datasource.password=password
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
```

EmployeeManagementSystem - EmployeeManagementSystemApplication [Spring Boot App] D:\vs code\jdk-21\bin\javaw.exe (13-Aug-2024, 11:12:45 pm) [pid: 16376]

```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/

 :: Spring Boot ::                (v3.3.2)

2024-08-13T23:12:48.393+05:30  INFO 16376 --- [           main] .e.E.EmployeeManagementSystemApplication : Starti
2024-08-13T23:12:48.397+05:30  INFO 16376 --- [           main] .e.E.EmployeeManagementSystemApplication : No act
2024-08-13T23:12:49.256+05:30  INFO 16376 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootst
2024-08-13T23:12:49.284+05:30  INFO 16376 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finish
2024-08-13T23:12:49.817+05:30  INFO 16376 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat
2024-08-13T23:12:49.831+05:30  INFO 16376 --- [           main] o.apache.catalina.core.StandardService   : Starti
2024-08-13T23:12:49.831+05:30  INFO 16376 --- [           main] o.apache.catalina.core.StandardEngine    : Starti
2024-08-13T23:12:49.905+05:30  INFO 16376 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initia
2024-08-13T23:12:49.905+05:30  INFO 16376 --- [           main] w.s.c.ServletWebServerApplicationContext : Root W
```



| | |
|---|---|
| Saved Settings: | Generic H2 (Embedded) |
| Setting Name: | Generic H2 (Embedded) |
| Driver Class: | org.h2.Driver |
| JDBC URL: | jdbc:h2:~/test |
| User Name: | sa |
| Password: | |



localhost:8081/h2-console/login.do?jsessionid=aead7abf27c461f30fa4921fc20dfb9b

Auto commit ✓ | Max rows: 1000 | Auto complete: Off | Auto select: On

jdbc:h2:mem:testdb
- DEPARTMENTS
- EMPLOYEES
- INFORMATION_SCHEMA
- Users

H2 2.2.224 (2023-09-17)

Run | Run Selected | Auto complete | Clear | SQL statement:

**Important Commands**

| | | |
|---|---|---|
| ? | | Displays this Help Page |
| | | Shows the Command History |
| | Ctrl+Enter | Executes the current SQL statement |
| | Shift+Enter | Executes the SQL statement defined by the text selection |
| | Ctrl+Space | Auto complete |
| | | Disconnects from the database |

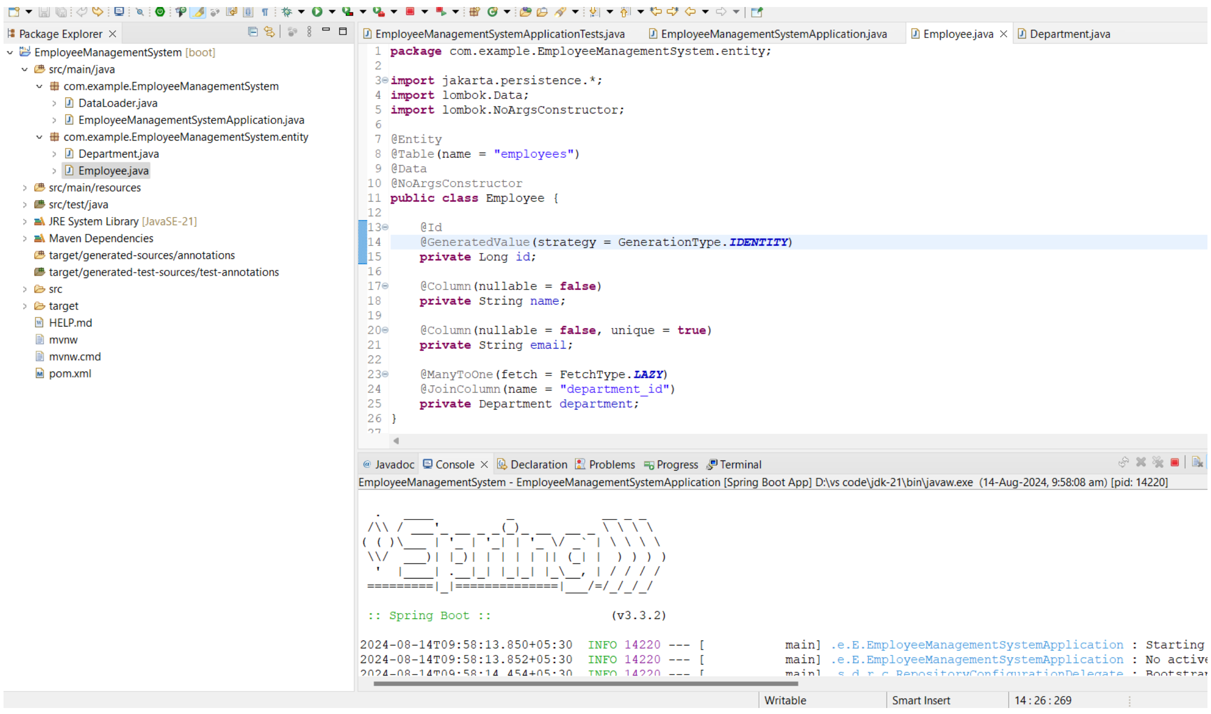
**Sample SQL Script**

| | |
|---|---|
| Delete the table if it exists | DROP TABLE IF EXISTS TEST; |
| Create a new table<br>with ID and NAME columns | CREATE TABLE TEST(ID INT PRIMARY KEY,<br>NAME VARCHAR(255)); |
| Add a new row | INSERT INTO TEST VALUES(1, 'Hello'); |
| Add another row | INSERT INTO TEST VALUES(2, 'World'); |
| Query the table | SELECT * FROM TEST ORDER BY ID; |
| Change data in a row | UPDATE TEST SET NAME='Hi' WHERE ID=1; |
| Remove a row | DELETE FROM TEST WHERE ID=2; |
| Help | HELP ... |

**Adding Database Drivers**

Additional database drivers can be registered by adding the Jar file location of the driver to the environment variables H2DRIVERS or CLASSPATH. E

## Exercise 2: Employee Management System - Creating Entities



Package Explorer
- EmployeeManagementSystem [boot]
  - src/main/java
    - com.example.EmployeeManagementSystem
      - DataLoader.java
      - EmployeeManagementSystemApplication.java
    - com.example.EmployeeManagementSystem.entity
      - Department.java
      - Employee.java
  - src/main/resources
  - src/test/java
  - JRE System Library [JavaSE-21]
  - Maven Dependencies
  - target/generated-sources/annotations
  - target/generated-test-sources/test-annotations
  - src
  - target
  - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml

EmployeeManagementSystemApplicationTests.java | EmployeeManagementSystemApplication.java | Employee.java | Department.java

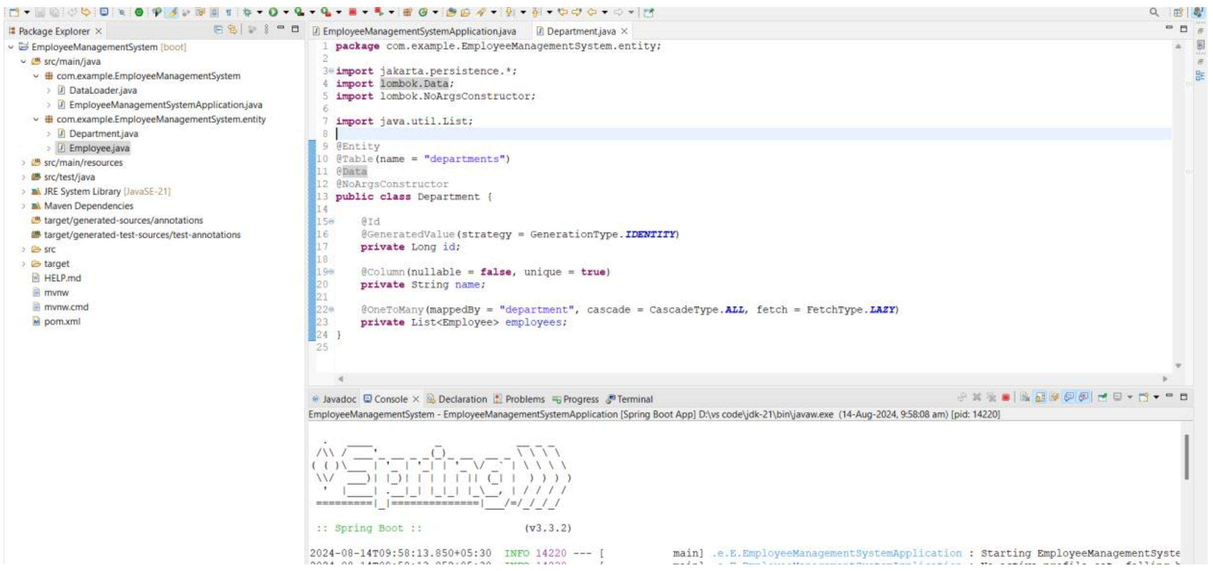
```java
package com.example.EmployeeManagementSystem.entity;

import jakarta.persistence.*;
import lombok.Data;
import lombok.NoArgsConstructor;

@Entity
@Table(name = "employees")
@Data
@NoArgsConstructor
public class Employee {

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

    @Column(nullable = false)
    private String name;

    @Column(nullable = false, unique = true)
    private String email;

    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "department_id")
    private Department department;
}
```

Javadoc | Console | Declaration | Problems | Progress | Terminal

EmployeeManagementSystem - EmployeeManagementSystemApplication [Spring Boot App] D:\vs code\jdk-21\bin\javaw.exe  (14-Aug-2024, 9:58:08 am) [pid: 14220]

```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/

 :: Spring Boot ::                (v3.3.2)

2024-08-14T09:58:13.850+05:30  INFO 14220 --- [           main] .e.E.EmployeeManagementSystemApplication : Starting
2024-08-14T09:58:13.852+05:30  INFO 14220 --- [           main] .e.E.EmployeeManagementSystemApplication : No active
2024-08-14T09:58:14.454+05:30  INFO 14220 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrap
```

Writable | Smart Insert | 14 : 26 : 269

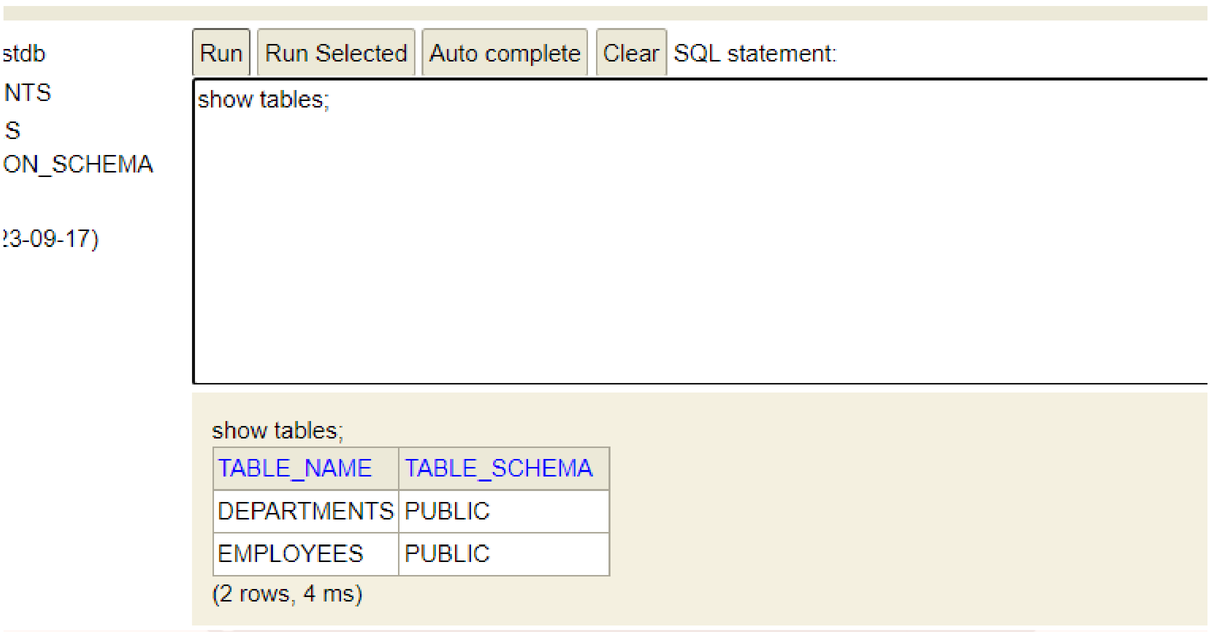


Package Explorer
EmployeeManagementSystem [boot]
src/main/java
com.example.EmployeeManagementSystem
DataLoader.java
EmployeeManagementSystemApplication.java
com.example.EmployeeManagementSystem.entity
Department.java
Employee.java
src/main/resources
src/test/java
JRE System Library [JavaSE-21]
Maven Dependencies
target/generated-sources/annotations
target/generated-test-sources/test-annotations
src
target
HELP.md
mvnw
mvnw.cmd
pom.xml
EmployeeManagementSystemApplication.java
Department.java
Javadoc
Console
Declaration
Problems
Progress
Terminal

```java
package com.example.EmployeeManagementSystem.entity;

import jakarta.persistence.*;
import lombok.Data;
import lombok.NoArgsConstructor;

import java.util.List;

@Entity
@Table(name = "departments")
@Data
@NoArgsConstructor
public class Department {

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

    @Column(nullable = false, unique = true)
    private String name;

    @OneToMany(mappedBy = "department", cascade = CascadeType.ALL, fetch = FetchType.LAZY)
    private List<Employee> employees;
}
```

EmployeeManagementSystem - EmployeeManagementSystemApplication [Spring Boot App] D:\vs code\jdk-21\bin\javaw.exe (14-Aug-2024, 9:58:08 am) [pid: 14220]

```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/

 :: Spring Boot ::                (v3.3.2)

2024-08-14T09:58:13.850+05:30  INFO 14220 --- [           main] .e.E.EmployeeManagementSystemApplication : Starting EmployeeManagementSyste
```

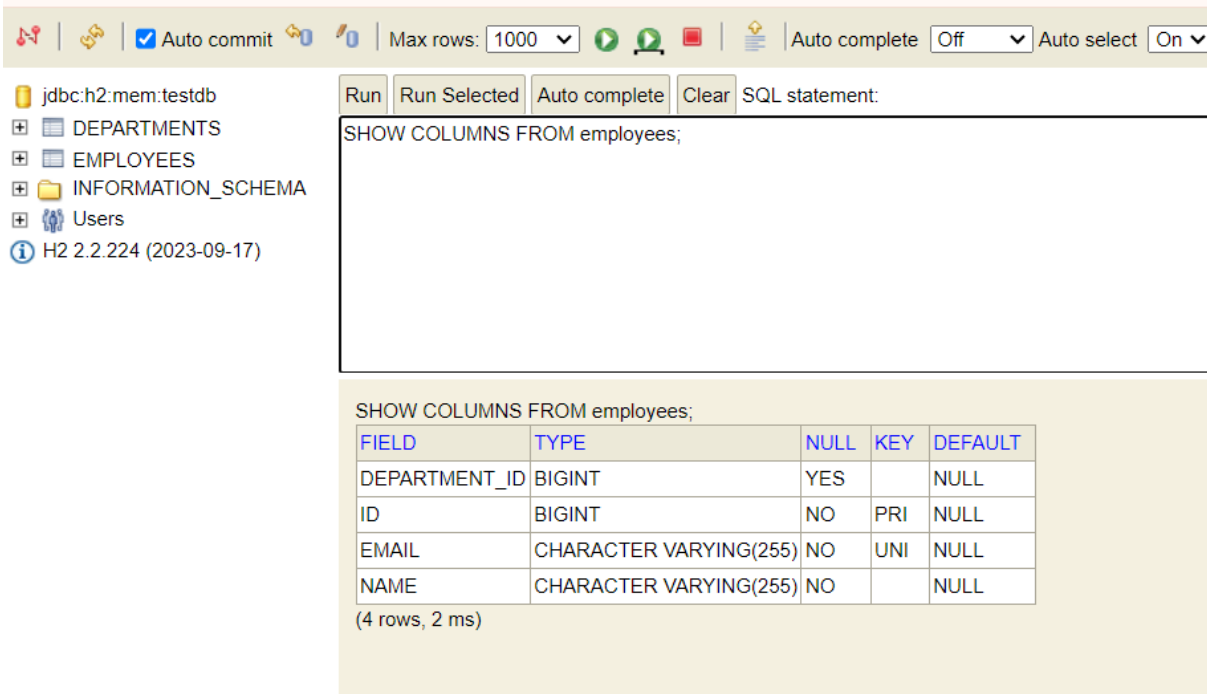


```
show tables;
```

show tables;

| TABLE_NAME | TABLE_SCHEMA |
|---|---|
| DEPARTMENTS | PUBLIC |
| EMPLOYEES | PUBLIC |

(2 rows, 4 ms)

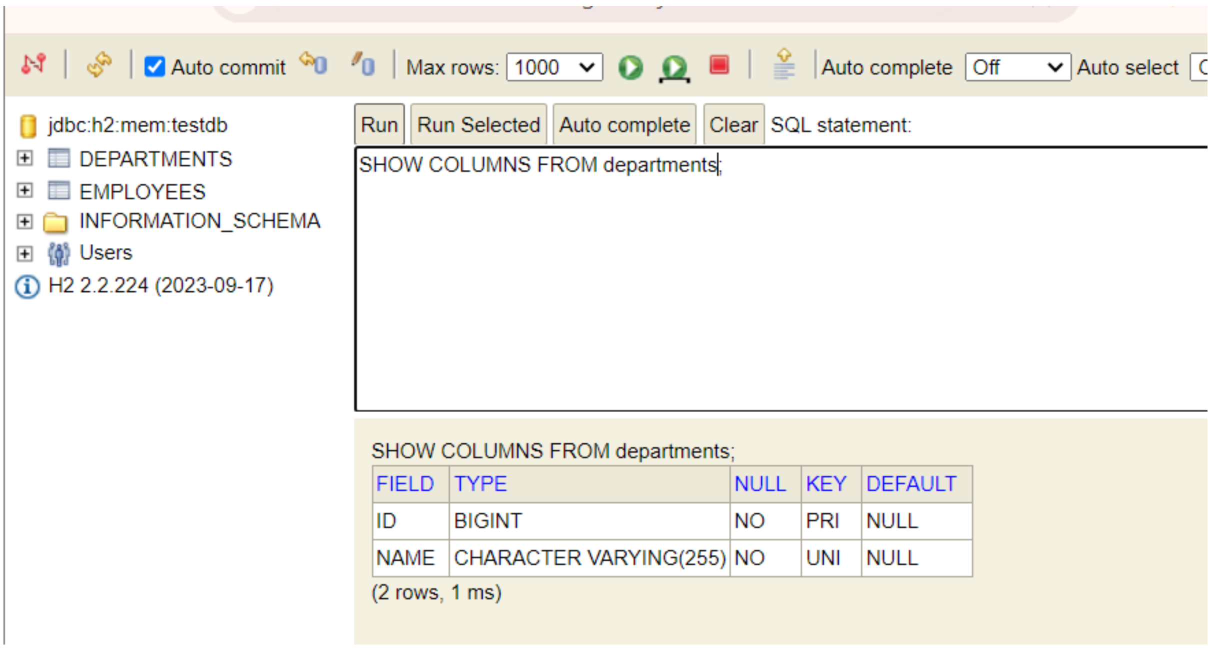


```
SHOW COLUMNS FROM employees;
```

SHOW COLUMNS FROM employees;

| FIELD | TYPE | NULL | KEY | DEFAULT |
|---|---|---|---|---|
| DEPARTMENT_ID | BIGINT | YES | | NULL |
| ID | BIGINT | NO | PRI | NULL |
| EMAIL | CHARACTER VARYING(255) | NO | UNI | NULL |
| NAME | CHARACTER VARYING(255) | NO | | NULL |

(4 rows, 2 ms)

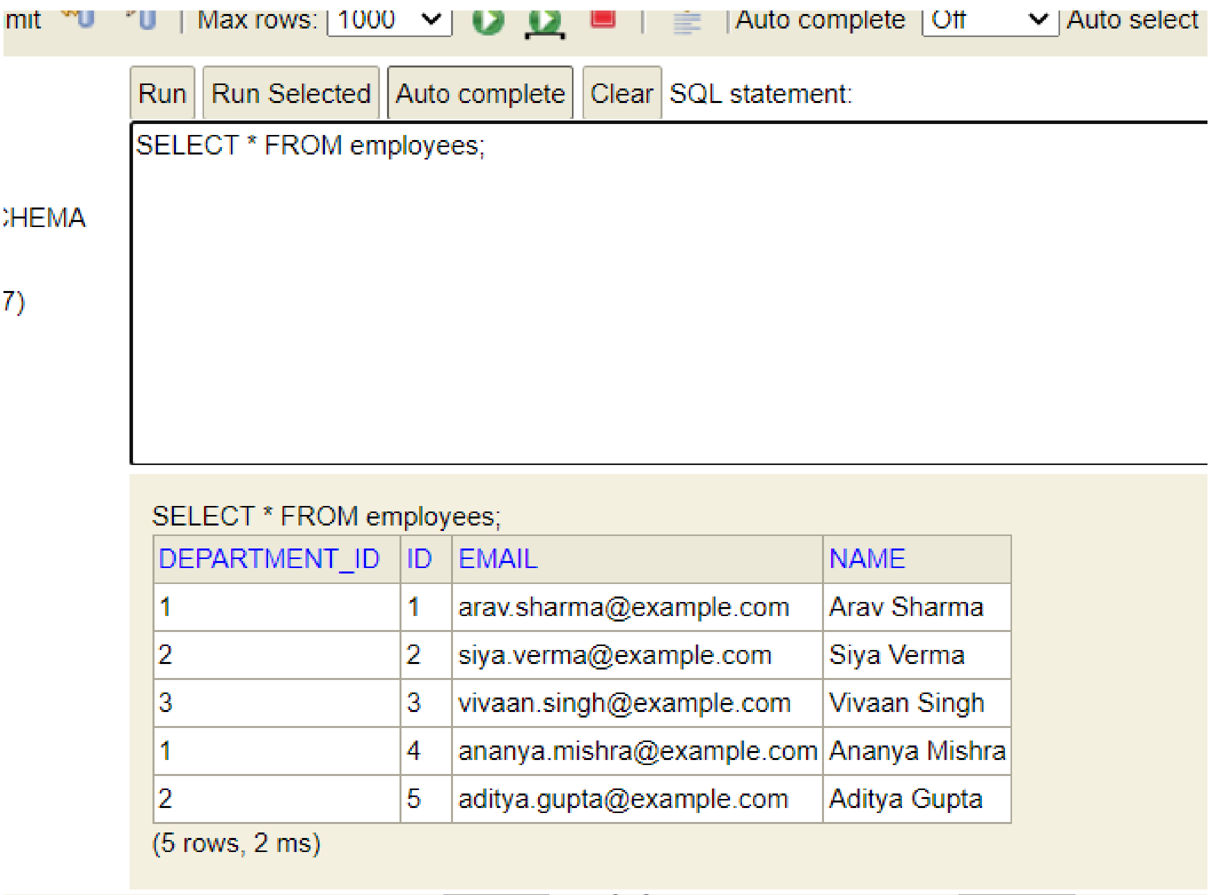


```
SHOW COLUMNS FROM departments;
```

SHOW COLUMNS FROM departments;

| FIELD | TYPE | NULL | KEY | DEFAULT |
|---|---|---|---|---|
| ID | BIGINT | NO | PRI | NULL |
| NAME | CHARACTER VARYING(255) | NO | UNI | NULL |

(2 rows, 1 ms)

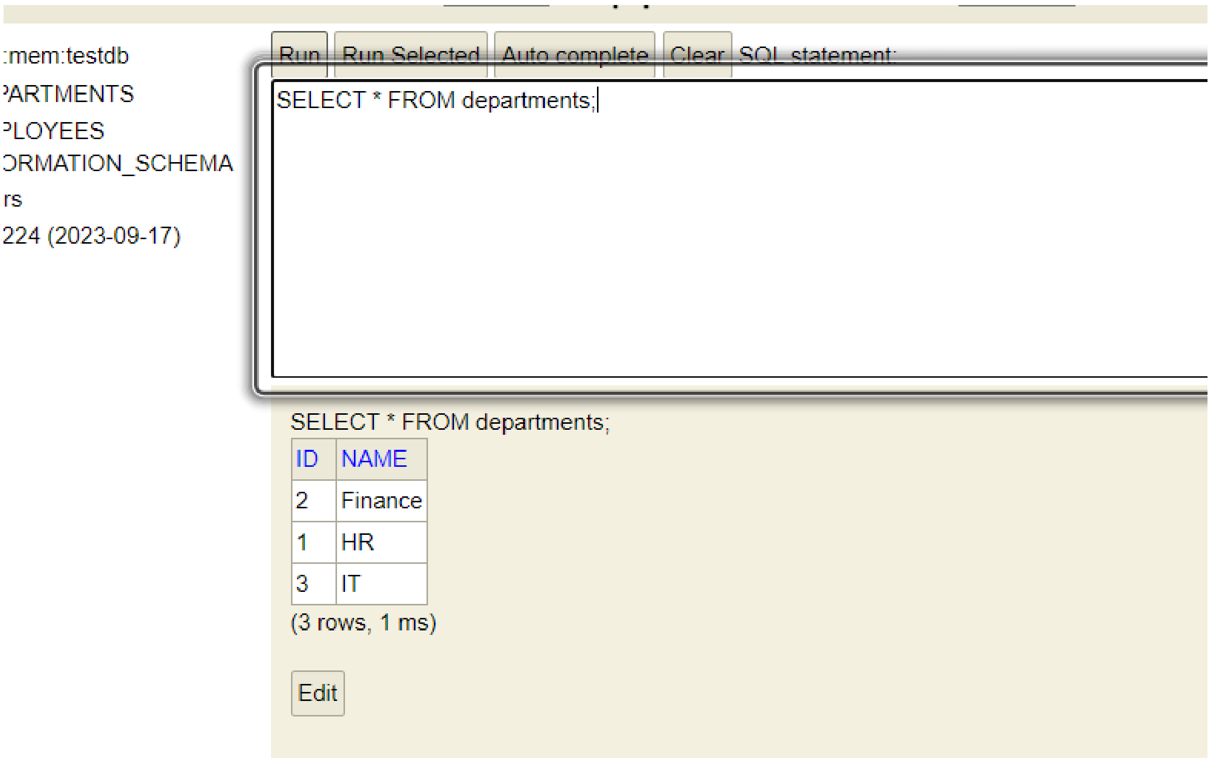


```
SELECT * FROM employees;
```

SELECT * FROM employees;

| DEPARTMENT_ID | ID | EMAIL | NAME |
|---|---|---|---|
| 1 | 1 | arav.sharma@example.com | Arav Sharma |
| 2 | 2 | siya.verma@example.com | Siya Verma |
| 3 | 3 | vivaan.singh@example.com | Vivaan Singh |
| 1 | 4 | ananya.mishra@example.com | Ananya Mishra |
| 2 | 5 | aditya.gupta@example.com | Aditya Gupta |

(5 rows, 2 ms)



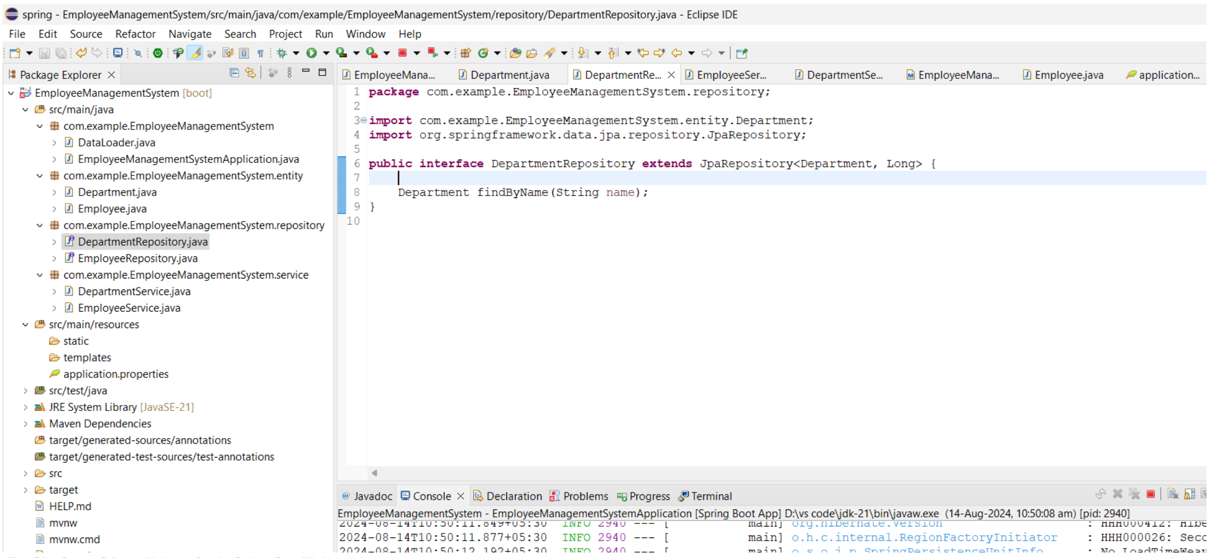
```
SELECT * FROM departments;
```

SELECT * FROM departments;

| ID | NAME |
|---|---|
| 2 | Finance |
| 1 | HR |
| 3 | IT |

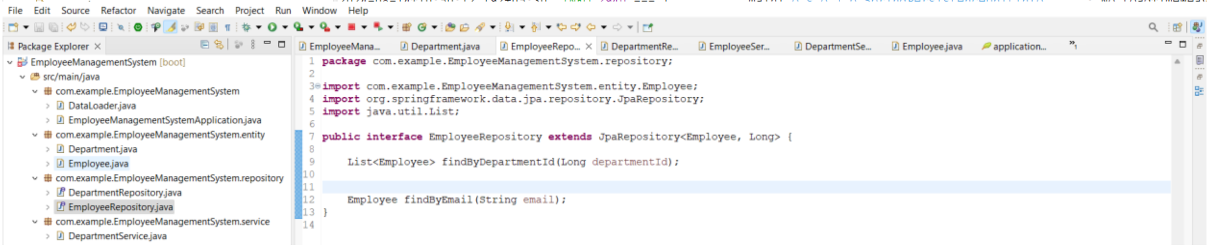
(3 rows, 1 ms)

## Exercise 3: Employee Management System - Creating Repositories

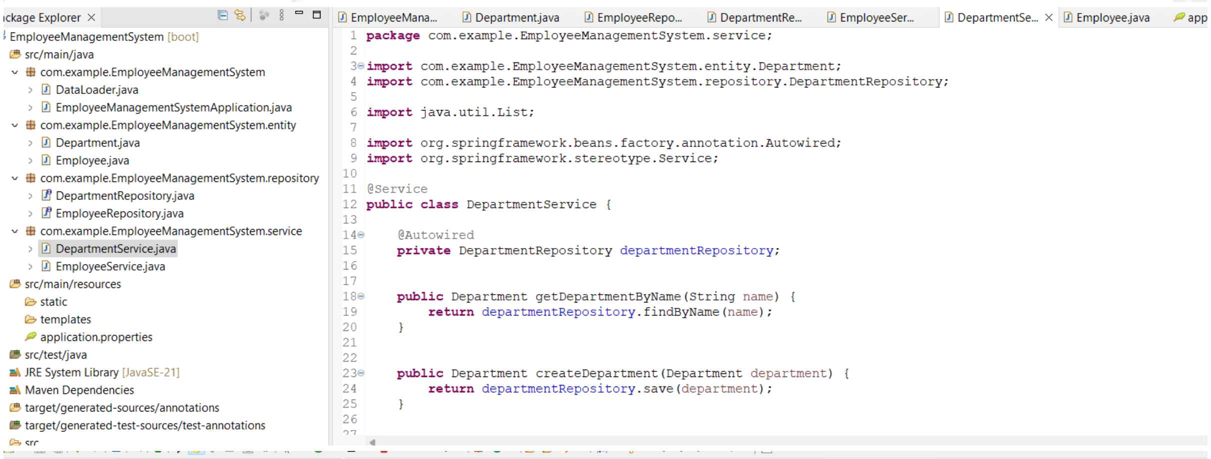

spring - EmployeeManagementSystem/src/main/java/com/example/EmployeeManagementSystem/repository/DepartmentRepository.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
EmployeeManagementSystem [boot]
src/main/java
com.example.EmployeeManagementSystem
DataLoader.java
EmployeeManagementSystemApplication.java
com.example.EmployeeManagementSystem.entity
Department.java
Employee.java
com.example.EmployeeManagementSystem.repository
DepartmentRepository.java
EmployeeRepository.java
com.example.EmployeeManagementSystem.service
DepartmentService.java
EmployeeService.java
src/main/resources
static
templates
application.properties
src/test/java
JRE System Library [JavaSE-21]
Maven Dependencies
target/generated-sources/annotations
target/generated-test-sources/test-annotations
src
target
HELP.md
mvnw
mvnw.cmd
Javadoc Console Declaration Problems Progress Terminal

```java
package com.example.EmployeeManagementSystem.repository;

import com.example.EmployeeManagementSystem.entity.Department;
import org.springframework.data.jpa.repository.JpaRepository;

public interface DepartmentRepository extends JpaRepository<Department, Long> {

    Department findByName(String name);
}
```

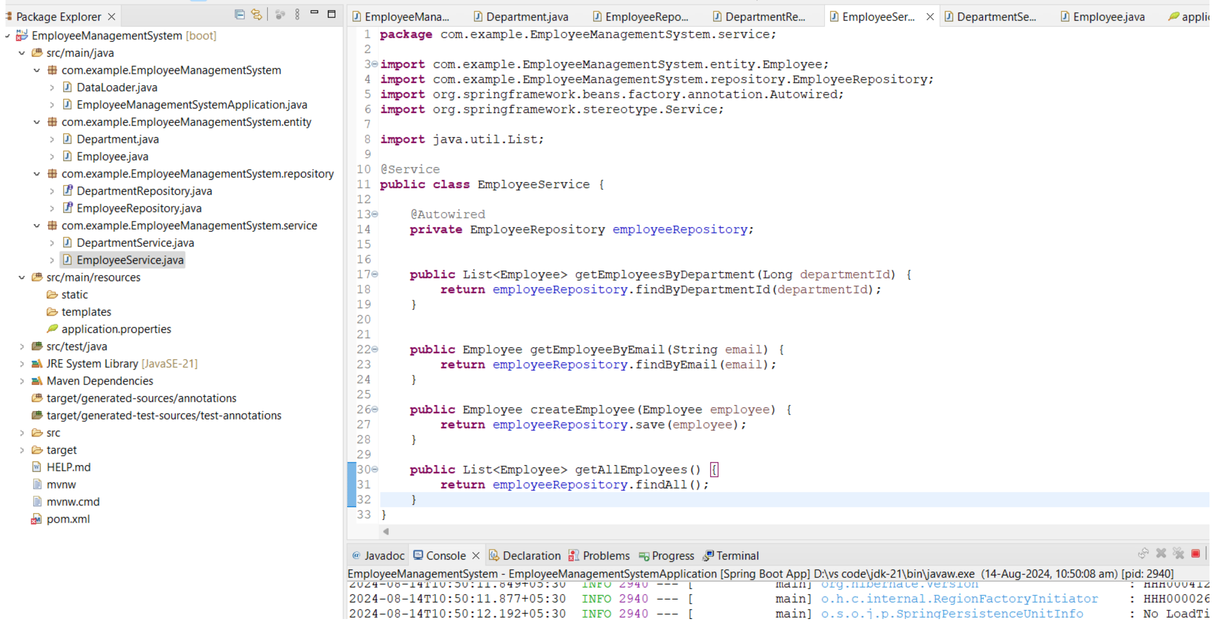
```
EmployeeManagementSystem - EmployeeManagementSystemApplication [Spring Boot App] D:\vs code\jdk-21\bin\javaw.exe  (14-Aug-2024, 10:50:08 am) [pid: 2940]
2024-08-14T10:50:11.877+05:30  INFO 2940 --- [           main] o.h.c.internal.RegionFactoryInitiator    : HHH000026: Secc
```


File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
EmployeeManagementSystem [boot]
src/main/java
com.example.EmployeeManagementSystem
DataLoader.java
EmployeeManagementSystemApplication.java
com.example.EmployeeManagementSystem.entity
Department.java
Employee.java
com.example.EmployeeManagementSystem.repository
DepartmentRepository.java
EmployeeRepository.java
com.example.EmployeeManagementSystem.service
DepartmentService.java

```java
package com.example.EmployeeManagementSystem.repository;

import com.example.EmployeeManagementSystem.entity.Employee;
import org.springframework.data.jpa.repository.JpaRepository;
import java.util.List;

public interface EmployeeRepository extends JpaRepository<Employee, Long> {

    List<Employee> findByDepartmentId(Long departmentId);


    Employee findByEmail(String email);
}
```


Package Explorer
EmployeeManagementSystem [boot]
src/main/java
com.example.EmployeeManagementSystem
DataLoader.java
EmployeeManagementSystemApplication.java
com.example.EmployeeManagementSystem.entity
Department.java
Employee.java
com.example.EmployeeManagementSystem.repository
DepartmentRepository.java
EmployeeRepository.java
com.example.EmployeeManagementSystem.service
DepartmentService.java
EmployeeService.java
src/main/resources
static
templates
application.properties
src/test/java
JRE System Library [JavaSE-21]
Maven Dependencies
target/generated-sources/annotations
target/generated-test-sources/test-annotations

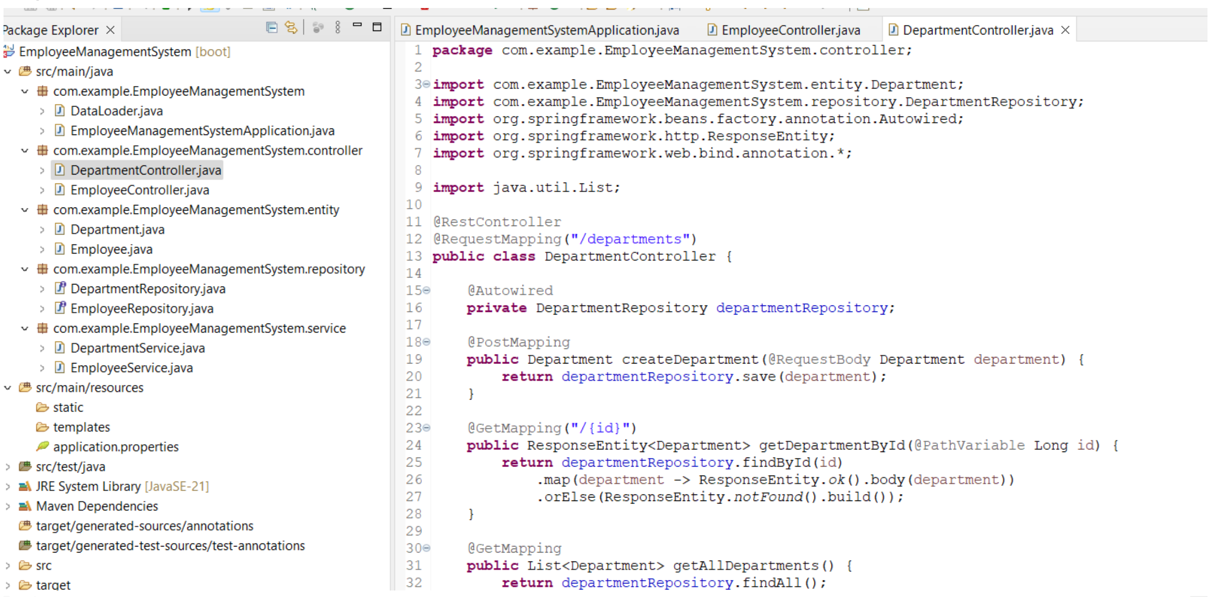
```java
package com.example.EmployeeManagementSystem.service;

import com.example.EmployeeManagementSystem.entity.Department;
import com.example.EmployeeManagementSystem.repository.DepartmentRepository;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class DepartmentService {

    @Autowired
    private DepartmentRepository departmentRepository;


    public Department getDepartmentByName(String name) {
        return departmentRepository.findByName(name);
    }


    public Department createDepartment(Department department) {
        return departmentRepository.save(department);
    }
```


Package Explorer
EmployeeManagementSystem [boot]
src/main/java
com.example.EmployeeManagementSystem
DataLoader.java
EmployeeManagementSystemApplication.java
com.example.EmployeeManagementSystem.entity
Department.java
Employee.java
com.example.EmployeeManagementSystem.repository
DepartmentRepository.java
EmployeeRepository.java
com.example.EmployeeManagementSystem.service
DepartmentService.java
EmployeeService.java
src/main/resources
static
templates
application.properties
src/test/java
JRE System Library [JavaSE-21]
Maven Dependencies
target/generated-sources/annotations
target/generated-test-sources/test-annotations
src
target
HELP.md
mvnw
mvnw.cmd
pom.xml
Javadoc Console Declaration Problems Progress Terminal

```java
package com.example.EmployeeManagementSystem.service;

import com.example.EmployeeManagementSystem.entity.Employee;
import com.example.EmployeeManagementSystem.repository.EmployeeRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class EmployeeService {

    @Autowired
    private EmployeeRepository employeeRepository;


    public List<Employee> getEmployeesByDepartment(Long departmentId) {
        return employeeRepository.findByDepartmentId(departmentId);
    }


    public Employee getEmployeeByEmail(String email) {
        return employeeRepository.findByEmail(email);
    }

    public Employee createEmployee(Employee employee) {
        return employeeRepository.save(employee);
    }

    public List<Employee> getAllEmployees() {
        return employeeRepository.findAll();
    }
}
```

```
EmployeeManagementSystem - EmployeeManagementSystemApplication [Spring Boot App] D:\vs code\jdk-21\bin\javaw.exe  (14-Aug-2024, 10:50:08 am) [pid: 2940]
2024-08-14T10:50:11.877+05:30  INFO 2940 --- [           main] o.h.c.internal.RegionFactoryInitiator    : HHH000026
2024-08-14T10:50:12.192+05:30  INFO 2940 --- [           main] o.s.o.j.p.SpringPersistenceUnitInfo      : No LoadTi
```

# Exercise 4: Employee Management System - Implementing CRUD Operations

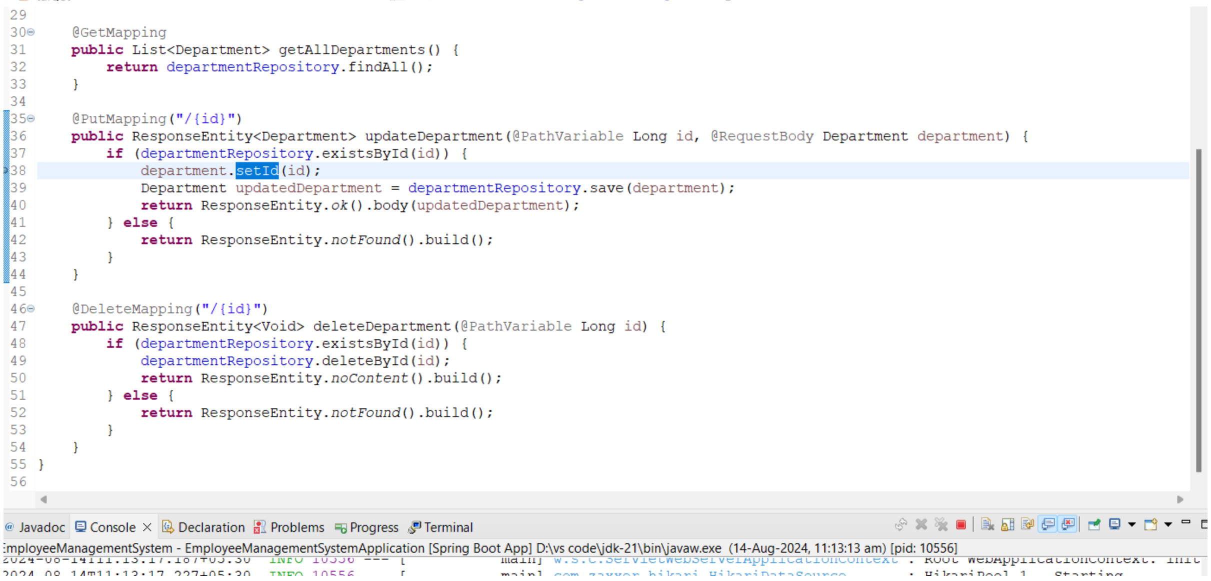


Package Explorer

```
EmployeeManagementSystem [boot]
  src/main/java
    com.example.EmployeeManagementSystem
      DataLoader.java
      EmployeeManagementSystemApplication.java
    com.example.EmployeeManagementSystem.controller
      DepartmentController.java
      EmployeeController.java
    com.example.EmployeeManagementSystem.entity
      Department.java
      Employee.java
    com.example.EmployeeManagementSystem.repository
      DepartmentRepository.java
      EmployeeRepository.java
    com.example.EmployeeManagementSystem.service
      DepartmentService.java
      EmployeeService.java
  src/main/resources
    static
    templates
    application.properties
  src/test/java
  JRE System Library [JavaSE-21]
  Maven Dependencies
  target/generated-sources/annotations
  target/generated-test-sources/test-annotations
  src
  target
```

EmployeeManagementSystemApplication.java | EmployeeController.java | DepartmentController.java

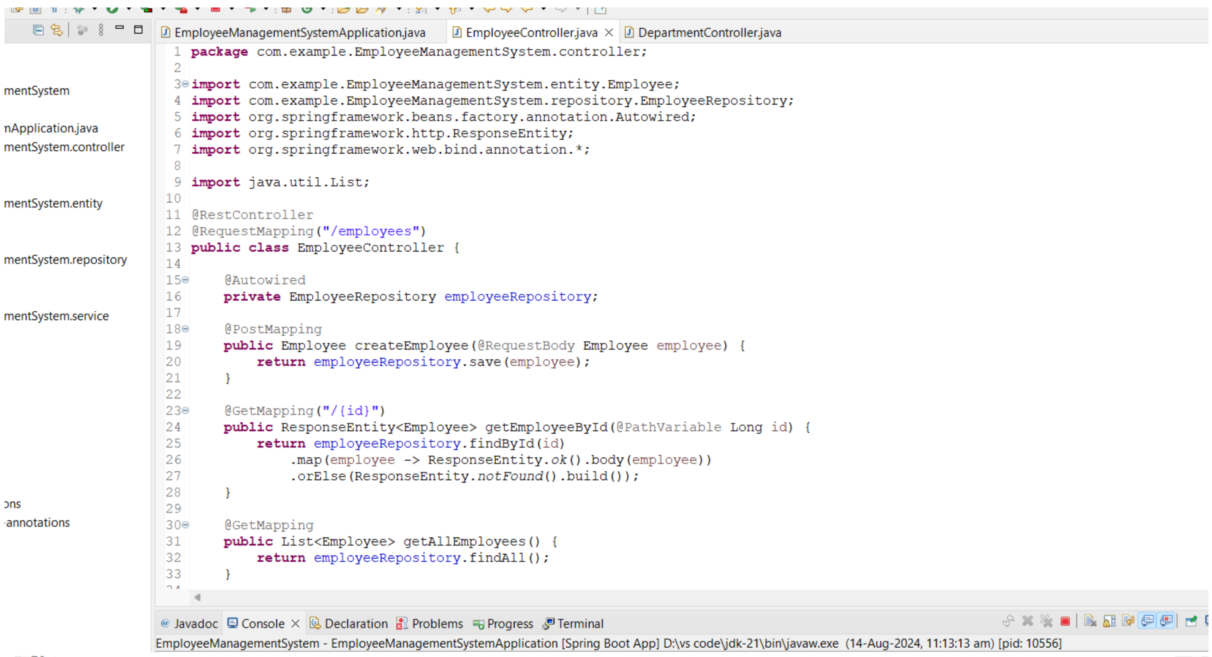
```java
package com.example.EmployeeManagementSystem.controller;

import com.example.EmployeeManagementSystem.entity.Department;
import com.example.EmployeeManagementSystem.repository.DepartmentRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
@RequestMapping("/departments")
public class DepartmentController {

    @Autowired
    private DepartmentRepository departmentRepository;

    @PostMapping
    public Department createDepartment(@RequestBody Department department) {
        return departmentRepository.save(department);
    }

    @GetMapping("/{id}")
    public ResponseEntity<Department> getDepartmentById(@PathVariable Long id) {
        return departmentRepository.findById(id)
            .map(department -> ResponseEntity.ok().body(department))
            .orElse(ResponseEntity.notFound().build());
    }

    @GetMapping
    public List<Department> getAllDepartments() {
        return departmentRepository.findAll();
```



```java

    @GetMapping
    public List<Department> getAllDepartments() {
        return departmentRepository.findAll();
    }

    @PutMapping("/{id}")
    public ResponseEntity<Department> updateDepartment(@PathVariable Long id, @RequestBody Department department) {
        if (departmentRepository.existsById(id)) {
            department.setId(id);
            Department updatedDepartment = departmentRepository.save(department);
            return ResponseEntity.ok().body(updatedDepartment);
        } else {
            return ResponseEntity.notFound().build();
        }
    }

    @DeleteMapping("/{id}")
    public ResponseEntity<Void> deleteDepartment(@PathVariable Long id) {
        if (departmentRepository.existsById(id)) {
            departmentRepository.deleteById(id);
            return ResponseEntity.noContent().build();
        } else {
            return ResponseEntity.notFound().build();
        }
    }
}

```

Javadoc | Console | Declaration | Problems | Progress | Terminal

```
EmployeeManagementSystem - EmployeeManagementSystemApplication [Spring Boot App] D:\vs code\jdk-21\bin\javaw.exe  (14-Aug-2024, 11:13:13 am) [pid: 10556]
2024-08-14T11:13:17.187+05:30  INFO 10556 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: init
2024-08-14T11:13:17.227+05:30  INFO 10556 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Starting
```



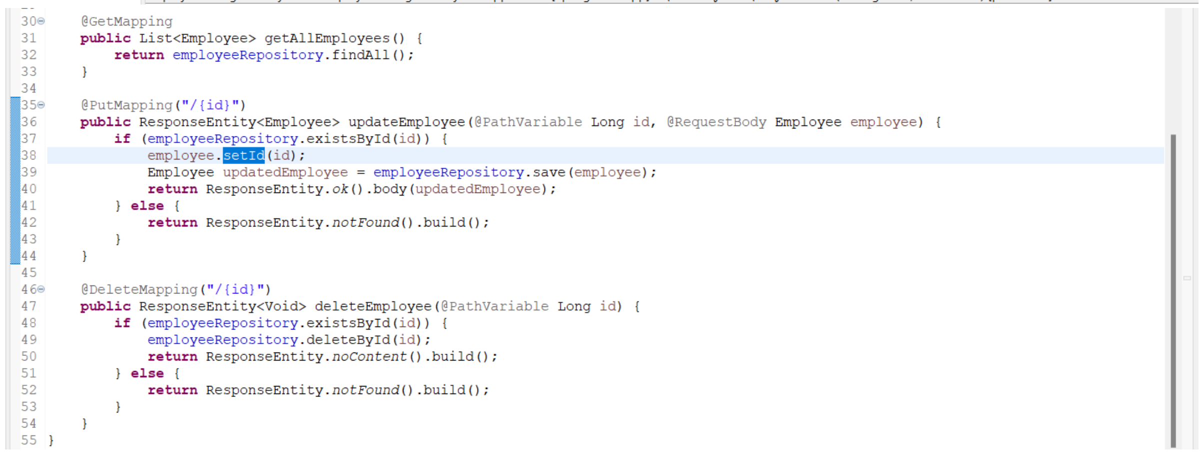
mentSystem
nApplication.java
mentSystem.controller
mentSystem.entity
mentSystem.repository
mentSystem.service
ons
annotations

EmployeeManagementSystemApplication.java | EmployeeController.java | DepartmentController.java

```java
package com.example.EmployeeManagementSystem.controller;

import com.example.EmployeeManagementSystem.entity.Employee;
import com.example.EmployeeManagementSystem.repository.EmployeeRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
@RequestMapping("/employees")
public class EmployeeController {

    @Autowired
    private EmployeeRepository employeeRepository;

    @PostMapping
    public Employee createEmployee(@RequestBody Employee employee) {
        return employeeRepository.save(employee);
    }

    @GetMapping("/{id}")
    public ResponseEntity<Employee> getEmployeeById(@PathVariable Long id) {
        return employeeRepository.findById(id)
            .map(employee -> ResponseEntity.ok().body(employee))
            .orElse(ResponseEntity.notFound().build());
    }

    @GetMapping
    public List<Employee> getAllEmployees() {
        return employeeRepository.findAll();
    }
```

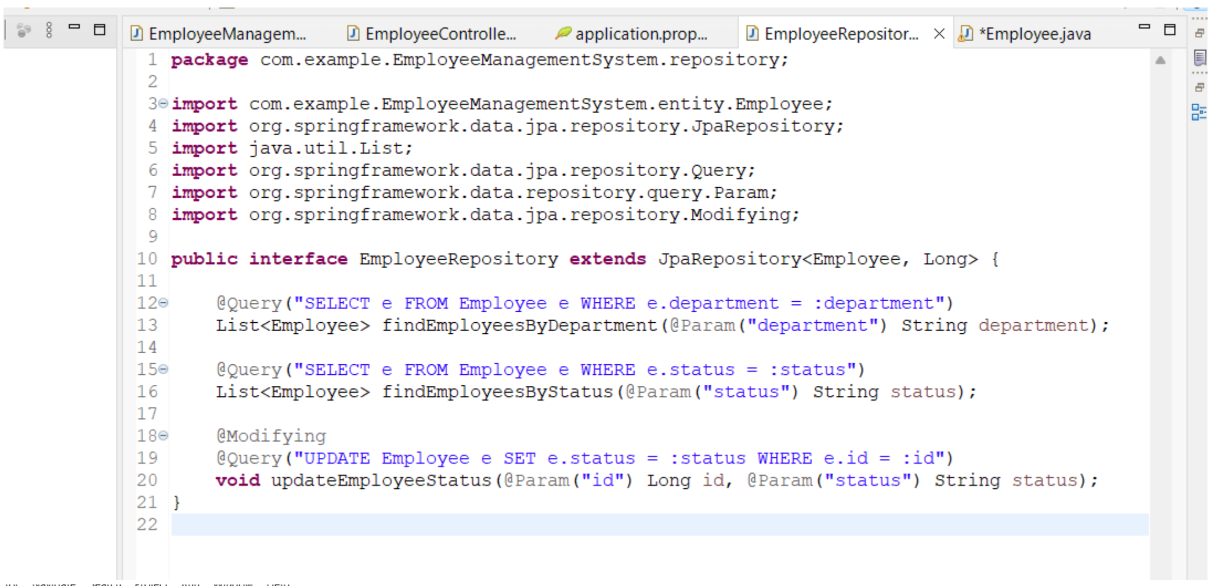
Javadoc | Console | Declaration | Problems | Progress | Terminal

```
EmployeeManagementSystem - EmployeeManagementSystemApplication [Spring Boot App] D:\vs code\jdk-21\bin\javaw.exe  (14-Aug-2024, 11:13:13 am) [pid: 10556]
```

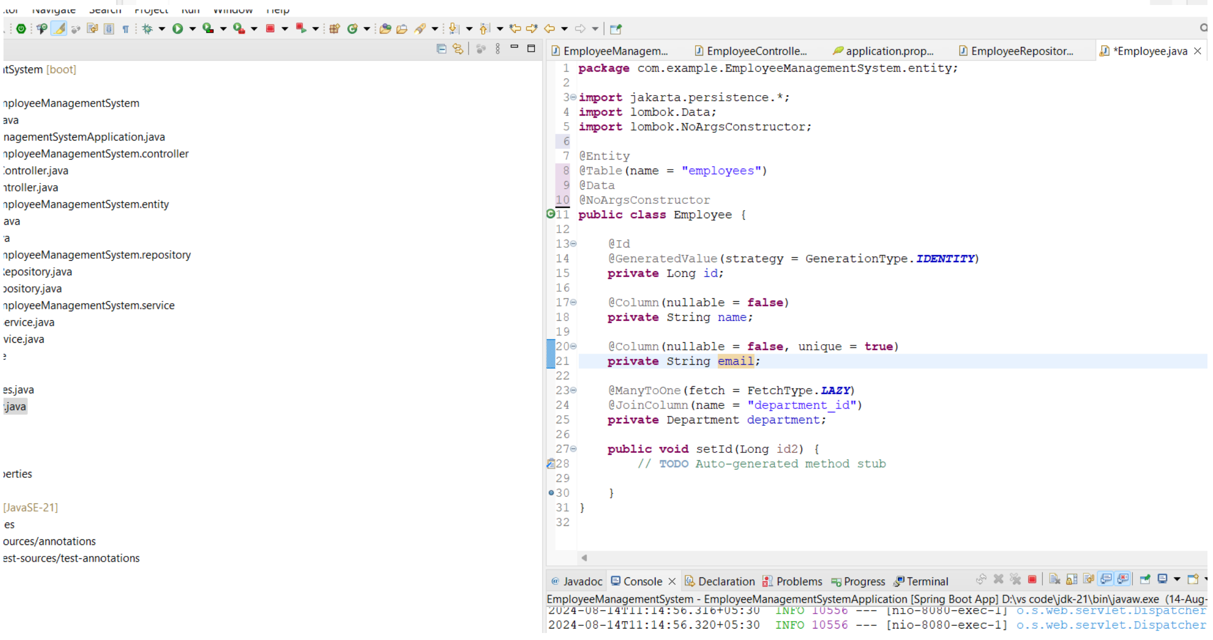


```java
    @GetMapping
    public List<Employee> getAllEmployees() {
        return employeeRepository.findAll();
    }

    @PutMapping("/{id}")
    public ResponseEntity<Employee> updateEmployee(@PathVariable Long id, @RequestBody Employee employee) {
        if (employeeRepository.existsById(id)) {
            employee.setId(id);
            Employee updatedEmployee = employeeRepository.save(employee);
            return ResponseEntity.ok().body(updatedEmployee);
        } else {
            return ResponseEntity.notFound().build();
        }
    }

    @DeleteMapping("/{id}")
    public ResponseEntity<Void> deleteEmployee(@PathVariable Long id) {
        if (employeeRepository.existsById(id)) {
            employeeRepository.deleteById(id);
            return ResponseEntity.noContent().build();
        } else {
            return ResponseEntity.notFound().build();
        }
    }
}
```

## Exercise 5: Employee Management System - Defining Query Methods



```java
package com.example.EmployeeManagementSystem.repository;

import com.example.EmployeeManagementSystem.entity.Employee;
import org.springframework.data.jpa.repository.JpaRepository;
import java.util.List;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.query.Param;
import org.springframework.data.jpa.repository.Modifying;

public interface EmployeeRepository extends JpaRepository<Employee, Long> {

    @Query("SELECT e FROM Employee e WHERE e.department = :department")
    List<Employee> findEmployeesByDepartment(@Param("department") String department);

    @Query("SELECT e FROM Employee e WHERE e.status = :status")
    List<Employee> findEmployeesByStatus(@Param("status") String status);

    @Modifying
    @Query("UPDATE Employee e SET e.status = :status WHERE e.id = :id")
    void updateEmployeeStatus(@Param("id") Long id, @Param("status") String status);
}

```



```
ıtSystem [boot]

ıployeeManagementSystem
ava
nagementSystemApplication.java
ıployeeManagementSystem.controller
ontroller.java
ıtroller.java
ıployeeManagementSystem.entity
ava
a
ıployeeManagementSystem.repository
epository.java
ository.java
ıployeeManagementSystem.service
ervice.java
vice.java
e

es.java
.java


erties

[JavaSE-21]
es
ources/annotations
est-sources/test-annotations
```

```java
package com.example.EmployeeManagementSystem.entity;

import jakarta.persistence.*;
import lombok.Data;
import lombok.NoArgsConstructor;

@Entity
@Table(name = "employees")
@Data
@NoArgsConstructor
public class Employee {

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

    @Column(nullable = false)
    private String name;

    @Column(nullable = false, unique = true)
    private String email;

    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "department_id")
    private Department department;

    public void setId(Long id2) {
        // TODO Auto-generated method stub

    }
}

```

```
EmployeeManagementSystem - EmployeeManagementSystemApplication [Spring Boot App] D:\vs code\jdk-21\bin\javaw.exe  (14-Aug-
2024-08-14T11:14:56.316+05:30  INFO 10556 --- [nio-8080-exec-1] o.s.web.servlet.Dispatcher
2024-08-14T11:14:56.320+05:30  INFO 10556 --- [nio-8080-exec-1] o.s.web.servlet.Dispatcher
```

(System hanged so couldn’t upload the screenshot)(updating the code )

Exercise 6: Employee Management System - Implementing Pagination and Sorting

EmployeeRepository:

```
import org.springframework.data.domain.Page;
import org.springframework.data.domain.Pageable;
import org.springframework.data.jpa.repository.JpaRepository;

public interface EmployeeRepository extends
JpaRepository<Employee, Long> {
    Page<Employee> findAll(Pageable pageable);
}
```

EmployeeService:

creating a method that takes Pageable as a parameter and returns the paginated Employee list.

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.data.domain.Page;
import org.springframework.data.domain.Pageable;
import org.springframework.stereotype.Service;

@Service
public class EmployeeService {

    @Autowired
    private EmployeeRepository employeeRepository;

    public Page<Employee> getEmployees(Pageable pageable) {
        return employeeRepository.findAll(pageable);
    }
}
```

Updating  EmployeeController to Handle Pagination:

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.data.domain.Page;
import org.springframework.data.domain.PageRequest;
import org.springframework.data.domain.Pageable;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class EmployeeController {

    @Autowired
    private EmployeeService employeeService;

    @GetMapping("/employees")
    public Page<Employee> getAllEmployees(
            @RequestParam(defaultValue = "0") int page,
            @RequestParam(defaultValue = "10") int size,
            @RequestParam(defaultValue = "id") String sortBy) {
        Pageable pageable = PageRequest.of(page, size,
Sort.by(sortBy));
        return employeeService.getEmployees(pageable);
    }
}
```

Testing: curl
http://localhost:8080/employees?page=0&size=5&sortBy=name

## Exercise 7: Employee Management System - Enabling Entity Auditing Business Scenario:

Enabling Auditing in the Main Application Class:

```
import org.springframework.boot.SpringApplication;
```

```
import
org.springframework.boot.autoconfigure.SpringBootApplication;
import
org.springframework.data.jpa.repository.config.EnableJpaAuditing;

@SpringBootApplication
@EnableJpaAuditing
public class EmployeeManagementSystemApplication {

    public static void main(String[] args) {

SpringApplication.run(EmployeeManagementSystemApplication.cla
ss, args);
    }
}
```

Adding Auditing Fields to Entities:

```
import org.springframework.data.annotation.CreatedBy;
import org.springframework.data.annotation.CreatedDate;
import org.springframework.data.annotation.LastModifiedBy;
import org.springframework.data.annotation.LastModifiedDate;
import
org.springframework.data.jpa.domain.support.AuditingEntityListen
er;

import javax.persistence.*;
import java.time.LocalDateTime;

@Entity
@EntityListeners(AuditingEntityListener.class)
public class Employee {

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

    private String name;
    private String email;

    @ManyToOne
    private Department department;

    @CreatedBy
    private String createdBy;

    @CreatedDate
    private LocalDateTime createdDate;

    @LastModifiedBy
    private String lastModifiedBy;

    @LastModifiedDate
    private LocalDateTime lastModifiedDate;

    // Getters and Setters
}
```
updating the Department entity:-
```
@Entity
@EntityListeners(AuditingEntityListener.class)
public class Department {

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

    private String name;
    @CreatedBy
    private String createdBy;

    @CreatedDate
    private LocalDateTime createdDate;

    @LastModifiedBy
    private String lastModifiedBy;

    @LastModifiedDate
    private LocalDateTime lastModifiedDate;

    // Getters and Setters
}
```

**Configur AuditorAware for User Information:**

```
import org.springframework.context.annotation.Bean;
import org.springframework.data.domain.AuditorAware;
import
org.springframework.data.jpa.repository.config.EnableJpaAuditing;
import org.springframework.stereotype.Component;

import java.util.Optional;

@Component
public class AuditorAwareImpl implements AuditorAware<String> {

    @Override
    public Optional<String> getCurrentAuditor() {
        // For simplicity, return a static username. Replace with actual
user context in a real application.
        return Optional.of("SystemUser");
    }
}
```

Registered AuditorAware in Configuration:

```
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.data.domain.AuditorAware;
import
org.springframework.data.jpa.repository.config.EnableJpaAuditing;

@Configuration
@EnableJpaAuditing(auditorAwareRef = "auditorAware")
public class AuditingConfig {

   @Bean
   public AuditorAware<String> auditorAware() {
      return new AuditorAwareImpl();
   }
}
```

## Exercise 8: Employee Management System - Creating Projections

Created an Interface for Employee Projection:

```
public interface EmployeeProjection {
   String getName();
   String getEmail();
   String getDepartmentName();
}
```

Modifying the EmployeeRepository::

```
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import java.util.List;

public interface EmployeeRepository extends JpaRepository<Employee, Long> {

   @Query("SELECT e.name AS name, e.email AS email, d.name AS departmentName " +
        "FROM Employee e JOIN e.department d")
  List<EmployeeProjection> findAllEmployeeProjections();
}
```

Class-Based Projections:

```
public class EmployeeDTO {

  private String name;
  private String email;
  private String departmentName;

  // Constructor
  public EmployeeDTO(String name, String email, String departmentName) {
    this.name = name;
    this.email = email;
    this.departmentName = departmentName;
  }
  // Getters and Setters
}
```

EmployeeRepository:

```
import org.springframework.data.jpa.repository.Query;
import org.springframework.stereotype.Repository;

import java.util.List;

@Repository
public interface EmployeeRepository extends JpaRepository<Employee, Long> {

  @Query("SELECT new com.example.EmployeeDTO(e.name, e.email, d.name) " +
        "FROM Employee e JOIN e.department d")
  List<EmployeeDTO> findAllEmployeeDTOs();
}
```

Use @Value Annotation for Custom Projections:

```
import org.springframework.beans.factory.annotation.Value;

public interface CustomEmployeeProjection {

  @Value("#{target.name + ' (' + target.department.name + ')'}")
  String getEmployeeWithDepartment();
}
```

Adding a method:

```
import org.springframework.data.jpa.repository.Query;
import org.springframework.stereotype.Repository;
import java.util.List;

@Repository
public interface EmployeeRepository extends JpaRepository<Employee, Long> {

   @Query("SELECT e FROM Employee e")
   List<CustomEmployeeProjection> findCustomEmployeeProjections();
}
```

## Exercise 9: Employee Management System - Customizing Data Source Configuration

**Externalize Configuration with application.properties**

```
# Primary Data Source Configuration
spring.datasource.primary.url=jdbc:mysql://localhost:3306/primarydb
spring.datasource.primary.username=root
spring.datasource.primary.password=root
spring.datasource.primary.driverClassName=com.mysql.cj.jdbc.Driver
spring.jpa.database-platform=org.hibernate.dialect.MySQL5Dialect
# Secondary Data Source Configuration
spring.datasource.secondary.url=jdbc:postgresql://localhost:5432/secondarydb
spring.datasource.secondary.username=postgres
spring.datasource.secondary.password=postgres
spring.datasource.secondary.driverClassName=org.postgresql.Driver

# application-dev.properties
spring.datasource.url=jdbc:h2:mem:devdb
spring.datasource.driverClassName=org.h2.Driver
spring.datasource.username=sa
spring.datasource.password=password
```

**Managing Multiple Data Sources:**

```
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.boot.autoconfigure.jdbc.DataSourceProperties;
import org.springframework.boot.context.properties.ConfigurationProperties;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.data.jpa.repository.config.EnableJpaRepositories;
import org.springframework.jdbc.datasource.DriverManagerDataSource;

import javax.sql.DataSource;

@Configuration
@EnableJpaRepositories(basePackages = "com.example.primary.repository",
      entityManagerFactoryRef = "primaryEntityManagerFactory",
      transactionManagerRef = "primaryTransactionManager")
```

```
public class DataSourceConfig {

  @Bean
  @ConfigurationProperties("spring.datasource.primary")
  public DataSourceProperties primaryDataSourceProperties() {
    return new DataSourceProperties();
  }

  @Bean
  public DataSource primaryDataSource() {
    return primaryDataSourceProperties().initializeDataSourceBuilder().build();
  }

  // Similarly, configure the secondary data source
}
```

Configure Entity Manager and Transaction Manager:

```
import org.springframework.orm.jpa.JpaTransactionManager;
import org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean;
import org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter;
import org.springframework.transaction.PlatformTransactionManager;

@Bean
public LocalContainerEntityManagerFactoryBean primaryEntityManagerFactory(
    @Qualifier("primaryDataSource") DataSource primaryDataSource) {
  LocalContainerEntityManagerFactoryBean em = new
LocalContainerEntityManagerFactoryBean();
  em.setDataSource(primaryDataSource);
  em.setPackagesToScan("com.example.primary.entity");
  em.setJpaVendorAdapter(new HibernateJpaVendorAdapter());
  return em;
}
@Bean
public PlatformTransactionManager primaryTransactionManager(
    @Qualifier("primaryEntityManagerFactory") LocalContainerEntityManagerFactoryBean
primaryEntityManagerFactory) {
  return new JpaTransactionManager(primaryEntityManagerFactory.getObject());
}
```

@Primary to Set a Default Data Source:

```
@Primary
@Bean
public DataSource primaryDataSource() {
  return primaryDataSourceProperties().initializeDataSourceBuilder().build();
}
```

**Exercise 10: Employee Management System - Hibernate-Specific Features**

**Hibernate-Specific Annotations**

```
import org.hibernate.annotations.Formula;
import org.hibernate.annotations.NaturalId;

@Entity
public class Employee {

   @NaturalId
   private String employeeCode;

   @Formula("(select count(e.id) from Employee e where
e.department_id = department_id)")
   private int employeeCount;
}
```

**using@Type to Customize Data Types:**

```
import org.hibernate.annotations.Type;

@Entity
public class Employee {
   @Type(type = "yes_no")
   private boolean active;

   // Other fields and methods
}
```

**Configuring Hibernate Dialect and Properties:**

```
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQ
L5Dialect

spring.jpa.properties.hibernate.show_sql=true
spring.jpa.properties.hibernate.format_sql=true
spring.jpa.properties.hibernate.use_sql_comments=true
spring.jpa.hibernate.ddl-auto=update
```

Implement Batch Processing

```
spring.jpa.properties.hibernate.jdbc.batch_size=20
spring.jpa.properties.hibernate.order_inserts=true
spring.jpa.properties.hibernate.order_updates=true
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;

@Service
public class EmployeeService {

  @Autowired
  private EmployeeRepository employeeRepository;

  @Transactional
  public void batchInsert(List<Employee> employees) {
    for (int i = 0; i < employees.size(); i++) {
      employeeRepository.save(employees.get(i));
      if (i % 20 == 0) { // Flush every 20 entities
        employeeRepository.flush();
        employeeRepository.clear();
      }
    }
  }
}
```