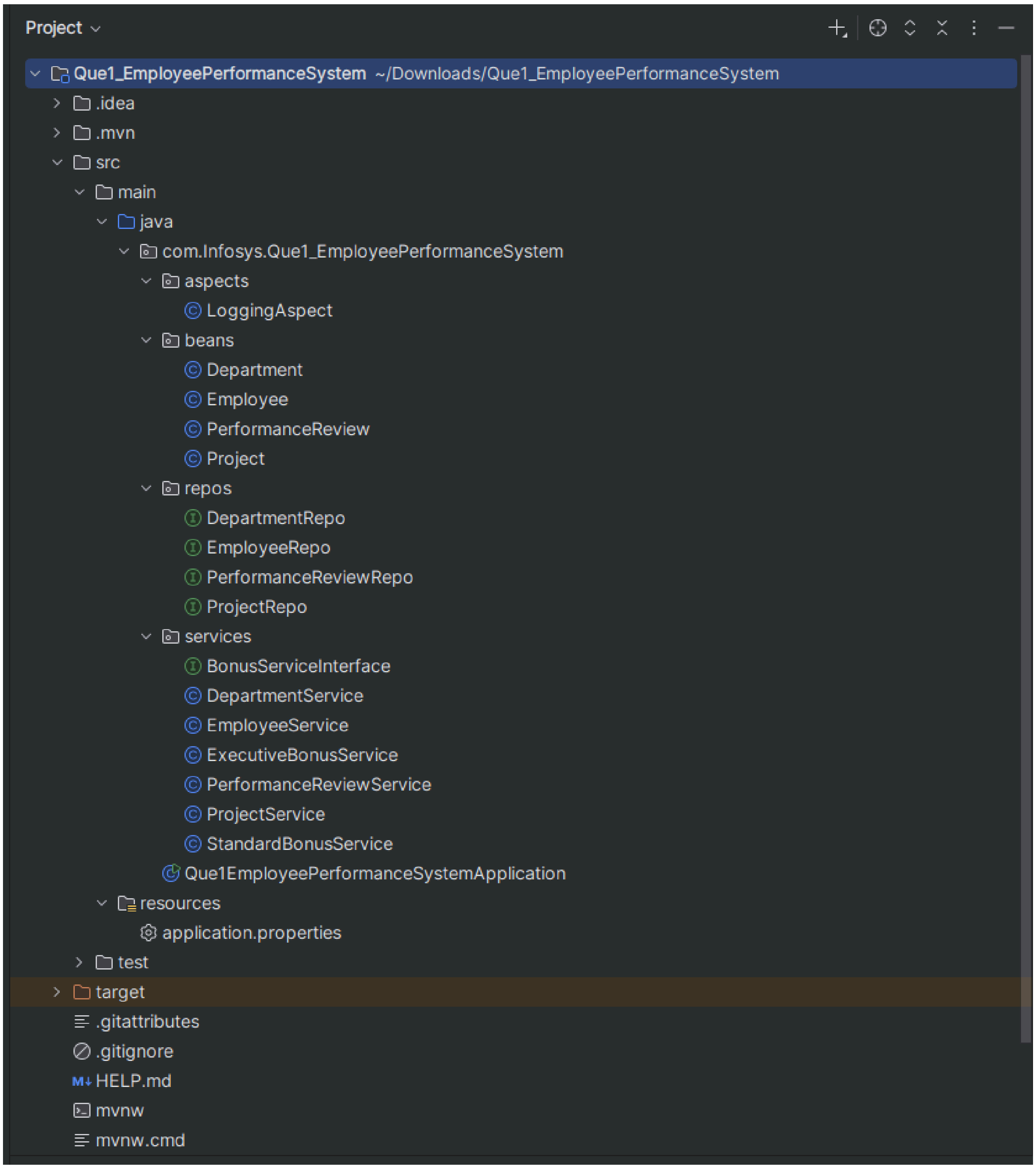
Spring I Capstone Project

Name - Aditya Pawar

USN - 72233061J

Employee Management System

Project Structure:



```
Project
Que1_EmployeePerformanceSystem ~/Downloads/Que1_EmployeePerformanceSystem
  .idea
  .mvn
  src
    main
      java
        com.Infosys.Que1_EmployeePerformanceSystem
          aspects
            LoggingAspect
          beans
            Department
            Employee
            PerformanceReview
            Project
          repos
            DepartmentRepo
            EmployeeRepo
            PerformanceReviewRepo
            ProjectRepo
          services
            BonusServiceInterface
            DepartmentService
            EmployeeService
            ExecutiveBonusService
            PerformanceReviewService
            ProjectService
            StandardBonusService
          Que1EmployeePerformanceSystemApplication
      resources
        application.properties
    test
  target
  .gitattributes
  .gitignore
  HELP.md
  mvnw
  mvnw.cmd
```

Application.properties:

```
spring.application.name=Que1_EmployeePerformanceSystem

spring.datasource.url=jdbc:postgresql://localhost:5432/capstoneProject
spring.datasource.username=postgres
spring.datasource.password=root
spring.jpa.database-platform=org.hibernate.dialect.PostgreSQLDialect
spring.jpa.hibernate.ddl-auto=create
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.format_sql=false

server.port=8080

spring.data.jpa.repositories.bootstrap-mode=default
spring.data.defer-datasource-initalization=true

#*********************************************
standard.multiplier=1000
executive.multiplier=2000
```

Code:

EmployeePerformanceSystem.java

```
package com.Infosys.Que1_EmployeePerformanceSystem;

import com.Infosys.Que1_EmployeePerformanceSystem.beans.Department;
import com.Infosys.Que1_EmployeePerformanceSystem.beans.Employee;
import com.Infosys.Que1_EmployeePerformanceSystem.beans.PerformanceR
import com.Infosys.Que1_EmployeePerformanceSystem.services.Department
import com.Infosys.Que1_EmployeePerformanceSystem.services.EmployeeSe
import com.Infosys.Que1_EmployeePerformanceSystem.services.ExecutiveBo
import com.Infosys.Que1_EmployeePerformanceSystem.services.StandardBo
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

import java.text.SimpleDateFormat;
import java.util.Arrays;
import java.util.Date;
import java.util.List;
import java.util.Scanner;

@SpringBootApplication
public class Que1EmployeePerformanceSystemApplication implements Comm
{

    public static void main(String[] args) {
        SpringApplication.run(Que1EmployeePerformanceSystemApplication.clas
    }

    @Autowired
    @Qualifier("standard")
    private StandardBonusService standardBonusService;

    @Autowired
    @Qualifier("executive")
    private ExecutiveBonusService executiveBonusService;

    @Autowired
    private DepartmentService departmentService;

    @Autowired
    private EmployeeService employeeService;

    @Autowired
    private PerformanceReviewService performanceReviewService;

    @Autowired
    private ProjectService projectService;

    private SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-

    @Override
    public void run(String... args) throws Exception {

        Scanner scanner = new Scanner(System.in);

        boolean running = true;
```

```
while (running) {

    System.out.println("\n======= EMPLOYEE PERFORMANCE SYSTEM =
    System.out.println("1. Add Department");
    System.out.println("2. Add Employee");
    System.out.println("3. Assign Projects");
    System.out.println("4. Add Performance Review");
    System.out.println("5. Calculate Bonus");
    System.out.println("6. Exit");
    System.out.println("======================================

    System.out.print("Enter your choice: ");
    int choice = scanner.nextInt();
    scanner.nextLine();

    switch (choice) {

        case 1:
            System.out.println("Enter Department Name: ");
            String departmentName = scanner.nextLine();

            Department department = new Department();
            department.setName(departmentName);

            departmentService.addDepartment(department);
            System.out.println("Department added successfully..." + departme
            break;

        case 2:
            System.out.println("Enter Employee Name: ");
            String employeeName = scanner.nextLine();
            System.out.println("Enter Employee Salary: ");
            double employeeSalary = scanner.nextDouble();
            scanner.nextLine();
            System.out.println("Enter Employee Rating (1-5): ");
            int employeeRating = scanner.nextInt();
            scanner.nextLine();
            System.out.println("Available Departments:");
            List<Department> departments = departmentService.getAllDepar
            for (Department dept : departments) {
                System.out.println(dept.getId() + ". " + dept.getName());
            }

            System.out.println("Select Department ID: ");
            int departmentId = scanner.nextInt();
            scanner.nextLine();

            Employee employee = new Employee();
            employee.setName(employeeName);
            employee.setSalary(employeeSalary);
            employee.setRating(employeeRating);

            Department department1 = new Department();
            department1.setId(departmentId);
            employee.setDepartment(department1);

            employeeService.addEmployee(employee);
            System.out.println("Employee added successfully...");
            break;

        case 3:
            System.out.println("Enter Employee ID: ");
            int empId = scanner.nextInt();
            scanner.nextLine();

            System.out.println("Enter Project IDs (comma-separated): ");
            String projectIds = scanner.nextLine();

            List<Integer> projectIdList = Arrays.stream(projectIds.split(","))
                    .map(String::trim)
                    .map(Integer::parseInt)
                    .toList();

            employeeService.assignProjects(empId, projectIdList);
            System.out.println("Projects assigned successfully to Employee ID
            break;

        case 4:
            System.out.println("Enter Employee ID for Performance Review: ")
            int empId2 = scanner.nextInt();
            scanner.nextLine();
            System.out.println("Enter Performance Rating (1-5): ");
            int performanceRating = scanner.nextInt();
            scanner.nextLine();
            System.out.println("Enter Review Date (YYYY-MM-DD): ");
            String date = scanner.nextLine();
            System.out.println("Enter Review Remarks: ");
            String remarks = scanner.nextLine();

            try {

                Date reviewDate = dateFormat.parse(date);

                PerformanceReview review =  new PerformanceReview();
                review.setRating(performanceRating);
                review.setReviewDate(reviewDate);
                review.setRemarks(remarks);

                employeeService.addPerformanceReview(empId2, review);
                System.out.println("Performance Review added successfully..."

            } catch (Exception e) {
                System.err.println("Invalid DateFormat (yyyy-mm-dd)..."+e.getM
            }
            break;

        case 5:

            System.out.println("Enter Employee Type (standard/executive): ");
            String type = scanner.nextLine().trim().toLowerCase();

            System.out.println("Enter Employee Rating (1-5): ");
```

}

```
                    int rating = scanner.nextInt();
                    scanner.nextLine();

                    double bonus;
                    try {

                        if ("standard".equals(type)) {
                            bonus = standardBonusService.calculateBonus(rating);
                        } else if ("executive".equals(type)) {
                            bonus = executiveBonusService.calculateBonus(rating);
                        } else {
                            System.out.println("Invalid employee type...");
                            return;
                        }

                        System.out.println("Calculated Bonus: " + bonus + " Rs.");
                    } catch (Exception e) {
                        System.err.println("Invalid input. Please enter a valid employee
                        type and rating..." + e.getMessage());
                    }

                    break;

                case 6:
                    System.out.println("Exiting system...Thank You For Visiting!!!");
                    running = false;
                    System.exit(0);
                    break;

                default:
                    System.err.println("Invalid choice. Please try again.");
                    break;

            }
        }
    }
}
```

Department.java

```
package com.Infosys.Que1_EmployeePerformanceSystem.beans;

import jakarta.persistence.*;
import jakarta.validation.constraints.NotBlank;

import java.util.List;

@Entity
@Table(name = "department")
public class Department {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;
    @NotBlank(message = "Department Name should be provided")
    private String name;

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

    // Getters and Setters
    public int getId() {
        return id;
    }

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

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
    public List<Employee> getEmployees() {
        return employees;
    }

    public void setEmployees(List<Employee> employees) {
        this.employees = employees;
    }

    @Override
    public String toString() {
        return "Department{" +
                "name='" + name + '\'' +
                ", id=" + id +
                '}';
    }
}
```

Employee.java

```
package com.Infosys.Que1_EmployeePerformanceSystem.beans;

import jakarta.persistence.*;
import jakarta.validation.constraints.NotBlank;

import java.util.ArrayList;
import java.util.List;

@Entity
@Table(name = "employee")
public class Employee {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;
    @NotBlank(message = "Employee name is mandatory")
```

```
private String name;
@NotBlank(message = "Salary should be provided")
private double salary;
@NotBlank(message = "Rating should be provided")
private int rating;

// Foreign key to Department
@ManyToOne
@JoinColumn(name = "department_id")
private Department department;

@OneToMany(mappedBy = "employee", cascade = CascadeType.ALL)
private List<PerformanceReview> performanceReviews;

@ManyToMany
@JoinTable(
        name = "employee_projects",
        joinColumns = @JoinColumn(name = "employee_id"),
        inverseJoinColumns = @JoinColumn(name = "project_id")
)
private List<Project> projects = new ArrayList<>();

public int getId() {
    return id;
}

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

public String getName() {
    return name;
}

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

```
    public double getSalary() {
        return salary;
    }

    public void setSalary(double salary) {
        this.salary = salary;
    }

    public int getRating() {
        return rating;
    }

    public void setRating(int rating) {
        this.rating = rating;
    }

    public Department getDepartment() {
        return department;
    }

    public void setDepartment(Department department) {
        this.department = department;
    }

    public List<PerformanceReview> getPerformanceReviews() {
        return performanceReviews;
    }

    public void setPerformanceReviews(List<PerformanceReview> performanc
        this.performanceReviews = performanceReviews;
    }

    public List<Project> getProjects() {
        return projects;
    }

    public void setProjects(List<Project> projects) {
        this.projects = projects;
    }

    @Override
    public String toString() {
        return "Employee{" +
                "id=" + id +
                ", name='" + name + '\'' +
                ", salary=" + salary +
                ", rating=" + rating +
                '}';
    }
}
```

PerformanceReview.java

```
package com.Infosys.Que1_EmployeePerformanceSystem.beans;

import jakarta.persistence.*;
import jakarta.validation.constraints.NotBlank;

import java.util.Date;

@Entity
@Table(name = "performance_review")
public class PerformanceReview {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;
    @NotBlank(message = "Rating should be provided")
    private int rating;
    @Column(name = "review_date")
    @NotBlank(message = "Review date should be provided in the Format (yyy
    private Date reviewDate;
    @NotBlank(message = "Remarks should be provided")
```

```
private String remarks;

// Foreign key to Employee
@ManyToOne
@JoinColumn(name = "employee_id")
private Employee employee;

public int getId() {
    return id;
}

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

public int getRating() {
    return rating;
}

public void setRating(int rating) {
    this.rating = rating;
}

public Date getReviewDate() {
    return reviewDate;
}

public void setReviewDate(Date reviewDate) {
    this.reviewDate = reviewDate;
}

public String getRemarks() {
    return remarks;
}

public void setRemarks(String remarks) {
    this.remarks = remarks;
}
```

```
    public Employee getEmployee() {
        return employee;
    }

    public void setEmployee(Employee employee) {
        this.employee = employee;
    }

    @Override
    public String toString() {
        return "PerformanceReview{" +
                "id=" + id +
                ", rating=" + rating +
                ", reviewDate='" + reviewDate + '\'' +
                ", remarks='" + remarks + '\'' +
                '}';
    }
}
```

Projects.java

```
package com.Infosys.Que1_EmployeePerformanceSystem.beans;

import jakarta.persistence.*;
import jakarta.validation.constraints.NotBlank;

import java.util.ArrayList;
import java.util.List;

@Entity
@Table(name = "project")
public class Project {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
```

```
private int id;
@NotBlank(message = "Project title should be provided")
private String title;
@NotBlank(message = "Project duration in months should be provided")
private int durationMonths;

@ManyToMany(mappedBy = "projects")
private List<Employee> employees = new ArrayList<>();

public int getId() {
    return id;
}

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

public String getTitle() {
    return title;
}

public void setTitle(String title) {
    this.title = title;
}

public int getDurationMonths() {
    return durationMonths;
}

public void setDurationMonths(int durationMonths) {
    this.durationMonths = durationMonths;
}

public List<Employee> getEmployees() {
    return employees;
}

public void setEmployees(List<Employee> employees) {
        this.employees = employees;
    }

    @Override
    public String toString() {
        return "Project{" +
                "id=" + id +
                ", title='" + title + '\'' +
                ", durationMonths=" + durationMonths +
                '}';
    }
}
```

DepartmentRepo.java

```
package com.Infosys.Que1_EmployeePerformanceSystem.repos;

import com.Infosys.Que1_EmployeePerformanceSystem.beans.Department;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface DepartmentRepo extends JpaRepository<Department,Integer
}
```

EmployeeRepo.java

```
package com.Infosys.Que1_EmployeePerformanceSystem.repos;

import com.Infosys.Que1_EmployeePerformanceSystem.beans.Employee;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
```

```
public interface EmployeeRepo extends JpaRepository<Employee,Integer> {
}
```

PerformanceReviewRepo.java

```
package com.Infosys.Que1_EmployeePerformanceSystem.repos;

import com.Infosys.Que1_EmployeePerformanceSystem.beans.PerformanceR
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface PerformanceReviewRepo extends JpaRepository<Performanc
Integer> {
}
```

ProjectsRepo.java

```
package com.Infosys.Que1_EmployeePerformanceSystem.repos;

import com.Infosys.Que1_EmployeePerformanceSystem.beans.Project;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface ProjectRepo extends JpaRepository<Project,Integer> {
}
```

DepartmentService.java

```
package com.Infosys.Que1_EmployeePerformanceSystem.services;

import com.Infosys.Que1_EmployeePerformanceSystem.beans.Department;
import com.Infosys.Que1_EmployeePerformanceSystem.repos.DepartmentRe
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class DepartmentService {

    @Autowired
    private DepartmentRepo departmentRepo;

    public void addDepartment(Department department) {
        departmentRepo.save(department);
    }

    public Department getDepartmentById(int id) {
        return departmentRepo.findById(id).orElse(null);
    }

    public List<Department> getAllDepartments() {
        return departmentRepo.findAll();
    }
}
```

EmployeeService.java

```
package com.Infosys.Que1_EmployeePerformanceSystem.services;

import com.Infosys.Que1_EmployeePerformanceSystem.beans.Employee;
import com.Infosys.Que1_EmployeePerformanceSystem.beans.PerformanceR
import com.Infosys.Que1_EmployeePerformanceSystem.beans.Project;
import com.Infosys.Que1_EmployeePerformanceSystem.repos.EmployeeRepo
import com.Infosys.Que1_EmployeePerformanceSystem.repos.PerformanceR
import com.Infosys.Que1_EmployeePerformanceSystem.repos.ProjectRepo;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import java.util.List;

@Service
public class EmployeeService {

    @Autowired
    private EmployeeRepo employeeRepo;

    @Autowired
    private ProjectRepo projectRepo;

    @Autowired
    private PerformanceReviewRepo performanceReviewRepo;

    // Adding employee
    public void addEmployee(Employee employee) {
        employeeRepo.save(employee);
    }

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

    public Employee getEmployeeById(int id) {
        return employeeRepo.findById(id).orElse(null);
    }

    public void assignProjects(int employeeId, List<Integer> projectIds) {
        Employee employee = employeeRepo.findById(employeeId).orElse(null);

        List<Project> projects = projectRepo.findAllById(projectIds);
        if (employee != null) {
            employee.getProjects().addAll(projects);
            employeeRepo.save(employee);
        } else {
            throw new RuntimeException("Employee not found");
        }
    }

    public void addPerformanceReview(int employeeId, PerformanceReview rev
        Employee employee = employeeRepo.findById(employeeId).orElse(null);
        if (employee != null) {
            review.setEmployee(employee);
            performanceReviewRepo.save(review);
        } else {
            throw new RuntimeException("Employee not found");
        }
    }

}
```

PerformanceReviewService.java

```
package com.Infosys.Que1_EmployeePerformanceSystem.services;

import com.Infosys.Que1_EmployeePerformanceSystem.beans.PerformanceR
import com.Infosys.Que1_EmployeePerformanceSystem.repos.PerformanceRe
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class PerformanceReviewService {

    @Autowired
    private PerformanceReviewRepo performanceReviewRepo;

    public void addPerformanceReview(PerformanceReview performanceRevie
        performanceReviewRepo.save(performanceReview);
    }

    public PerformanceReview getPerformanceReviewById(int id) {
        return performanceReviewRepo.findById(id).orElse(null);
    }

}
```

ProjectsService.java

```
package com.Infosys.Que1_EmployeePerformanceSystem.services;

import com.Infosys.Que1_EmployeePerformanceSystem.beans.Project;
import com.Infosys.Que1_EmployeePerformanceSystem.repos.ProjectRepo;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class ProjectService {

    @Autowired
    private ProjectRepo projectRepo;

    public void addProject(Project project) {
        projectRepo.save(project);
    }
    public Project getProjectById(int id) {
        return projectRepo.findById(id).orElse(null);
    }

}
```

BonusServiceInterface.java

```
package com.Infosys.Que1_EmployeePerformanceSystem.services;

public interface BonusServiceInterface {
    double calculateBonus(int rating);
}
```

StandardBonusService.java

```
package com.Infosys.Que1_EmployeePerformanceSystem.services;

import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Component;

@Component
@Qualifier("standard")
public class StandardBonusService implements BonusServiceInterface{

    @Value("${standard.multiplier}")
    private double multiplier;

    @Override
    public double calculateBonus(int rating) {

        if (rating < 1 || rating > 5) {
            throw new IllegalArgumentException("Rating must be between 1 and 5'
        }
        return rating * multiplier;
    }
}
```

ExecutiveBonusService.java

```
package com.Infosys.Que1_EmployeePerformanceSystem.services;

import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Component;

@Component
@Qualifier("executive")
```

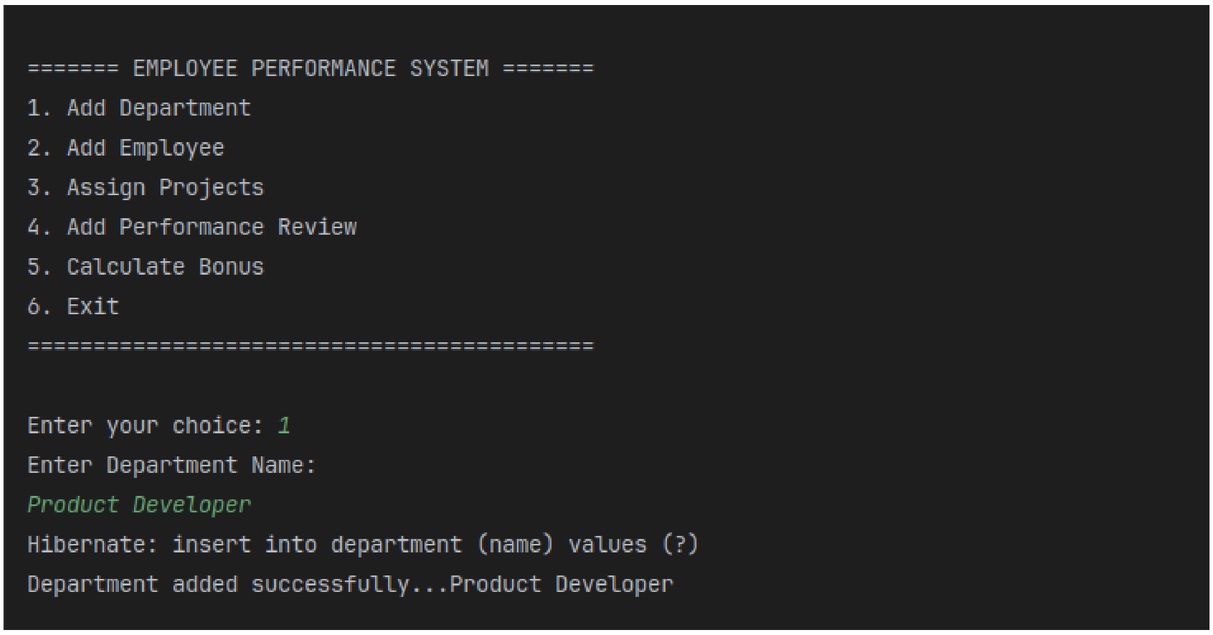
```
public class ExecutiveBonusService implements BonusServiceInterface{

    @Value("${executive.multiplier}")
    private double multiplier;

    @Override
    public double calculateBonus(int rating) {
        if (rating < 1 || rating > 5) {
            throw new IllegalArgumentException("Rating must be between 1 and 5'
        }
        return rating * multiplier;
    }
}
```

LoggerAspect.java

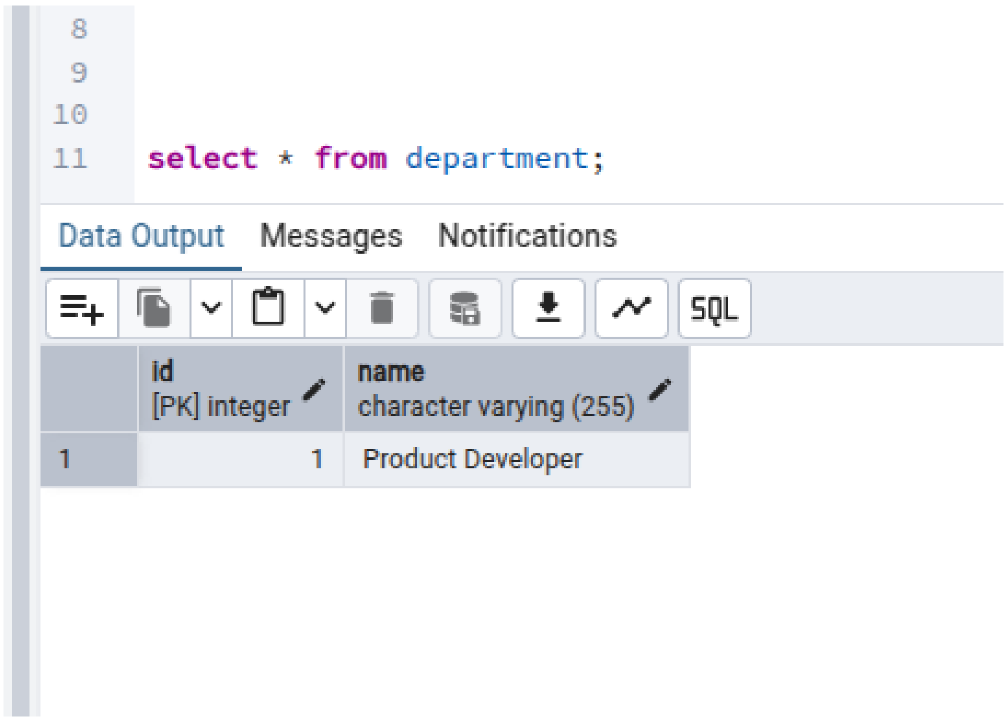
```
package com.Infosys.Que1_EmployeePerformanceSystem.aspects;

import org.aspectj.lang.ProceedingJoinPoint;
import org.aspectj.lang.annotation.Around;
import org.aspectj.lang.annotation.Aspect;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.stereotype.Component;

@Aspect
@Component
public class LoggingAspect {

    private final Logger logger = LoggerFactory.getLogger(this.getClass());

    @Around("execution(* com.example.Capstone.service.*.*(..))")
    public Object logServiceMethods(ProceedingJoinPoint joinPoint) throws
    Throwable {
        long startTime = System.currentTimeMillis();

        String methodName = joinPoint.getSignature().getName();

        Object[] args = joinPoint.getArgs();

        logger.info("Entering method: {} with arguments: {}", methodName, args)

        Object result = joinPoint.proceed();

        long endTime = System.currentTimeMillis();
        long executionTime = endTime - startTime;

        logger.info("Exiting method: {} with result: {}", methodName, result);
        logger.info("Execution time of method {}: {} ms", methodName,
         executionTime);

        return result;
    }
}
```

Output:

1. Add Department



```
======= EMPLOYEE PERFORMANCE SYSTEM =======
1. Add Department
2. Add Employee
3. Assign Projects
4. Add Performance Review
5. Calculate Bonus
6. Exit
===========================================

Enter your choice: 1
Enter Department Name:
Product Developer
Hibernate: insert into department (name) values (?)
Department added successfully...Product Developer
```



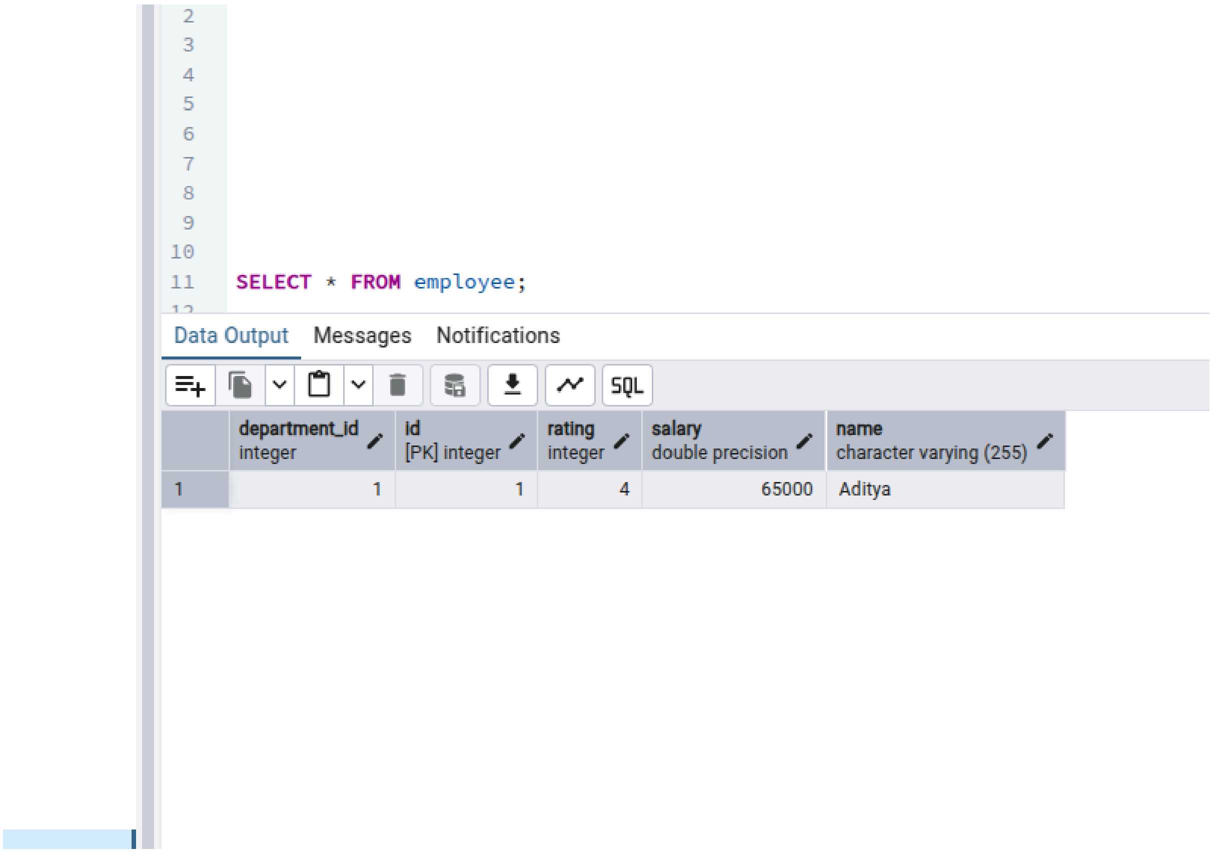
```
select * from department;
```

Data Output Messages Notifications

	id [PK] integer	name character varying (255)
1	1	Product Developer

2. Add Employee

```
======= EMPLOYEE PERFORMANCE SYSTEM =======
1. Add Department
2. Add Employee
3. Assign Projects
4. Add Performance Review
5. Calculate Bonus
6. Exit
===========================================

Enter your choice: 2
Enter Employee Name:
Aditya
Enter Employee Salary:
65000
Enter Employee Rating (1-5):
4
Available Departments:
Hibernate: select d1_0.id,d1_0.name from department d1_0
1. Product Developer
Select Department ID:
1
Hibernate: insert into employee (department_id,name,rating,salary) values (?,?,?,?)
Employee added successfully...
```

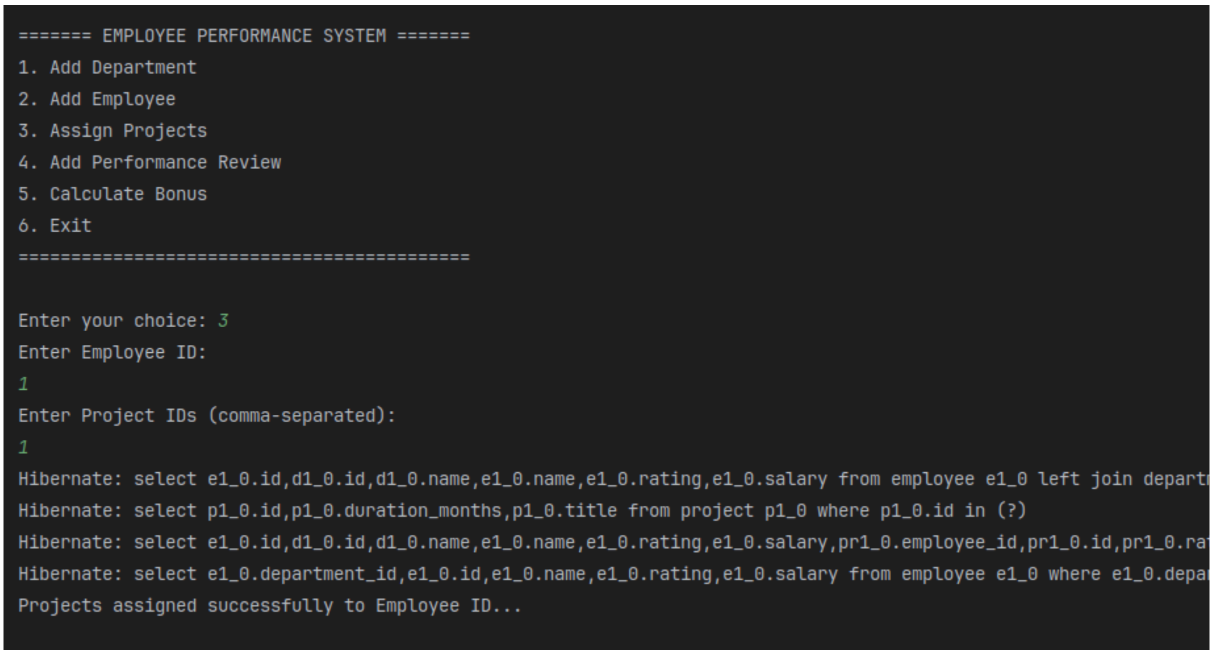


```sql
SELECT * FROM employee;
```

Data Output | Messages | Notifications

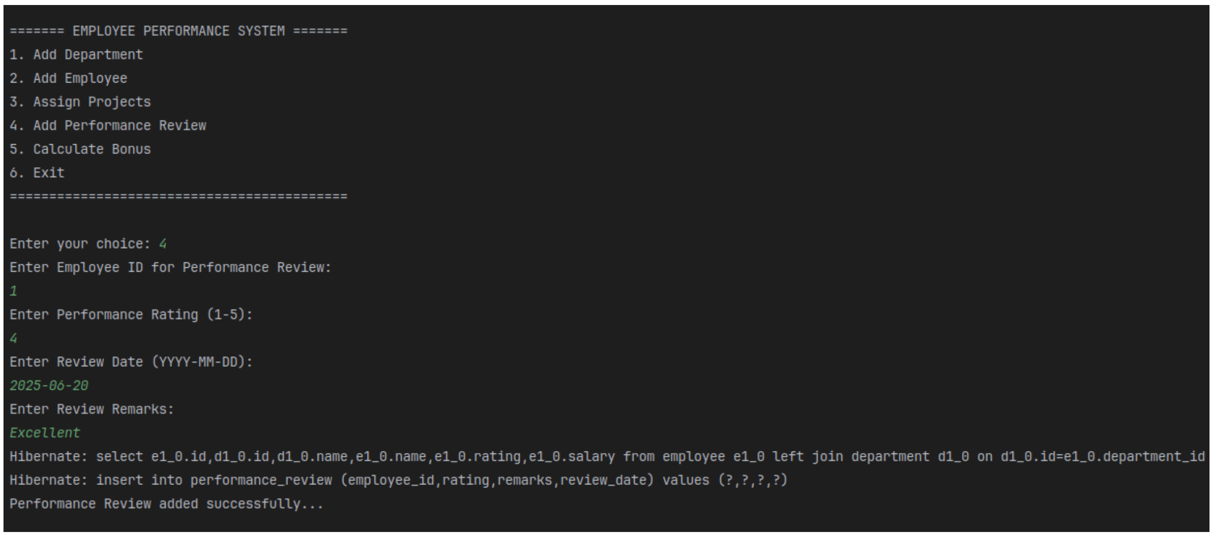
	department_id integer	id [PK] integer	rating integer	salary double precision	name character varying (255)
1	1	1	4	65000	Aditya

3. Assign Project

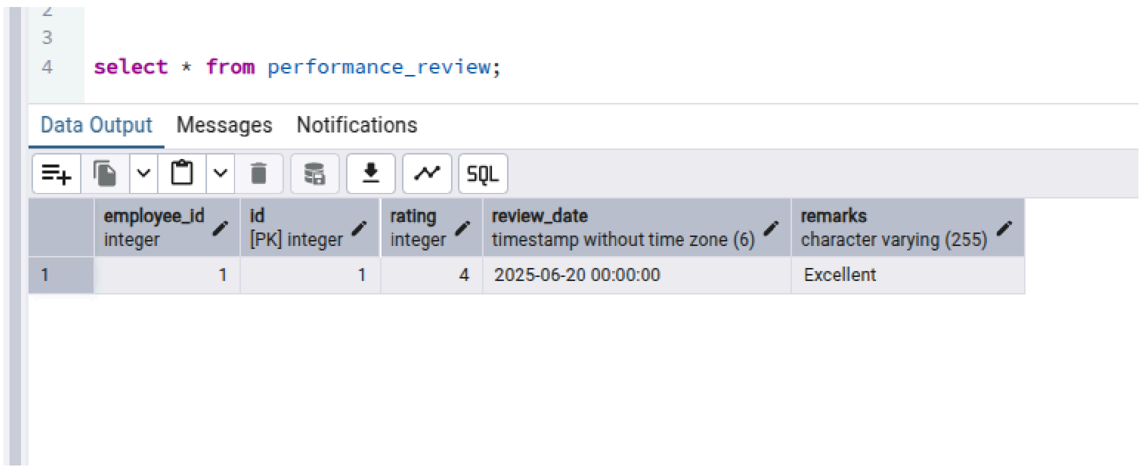


```
======= EMPLOYEE PERFORMANCE SYSTEM =======
1. Add Department
2. Add Employee
3. Assign Projects
4. Add Performance Review
5. Calculate Bonus
6. Exit
===========================================

Enter your choice: 3
Enter Employee ID:
1
Enter Project IDs (comma-separated):
1
Hibernate: select e1_0.id,d1_0.id,d1_0.name,e1_0.name,e1_0.rating,e1_0.salary from employee e1_0 left join depart
Hibernate: select p1_0.id,p1_0.duration_months,p1_0.title from project p1_0 where p1_0.id in (?)
Hibernate: select e1_0.id,d1_0.id,d1_0.name,e1_0.name,e1_0.rating,e1_0.salary,pr1_0.employee_id,pr1_0.id,pr1_0.ra
Hibernate: select e1_0.department_id,e1_0.id,e1_0.name,e1_0.rating,e1_0.salary from employee e1_0 where e1_0.depa
Projects assigned successfully to Employee ID...
```

4. Add Performance Review



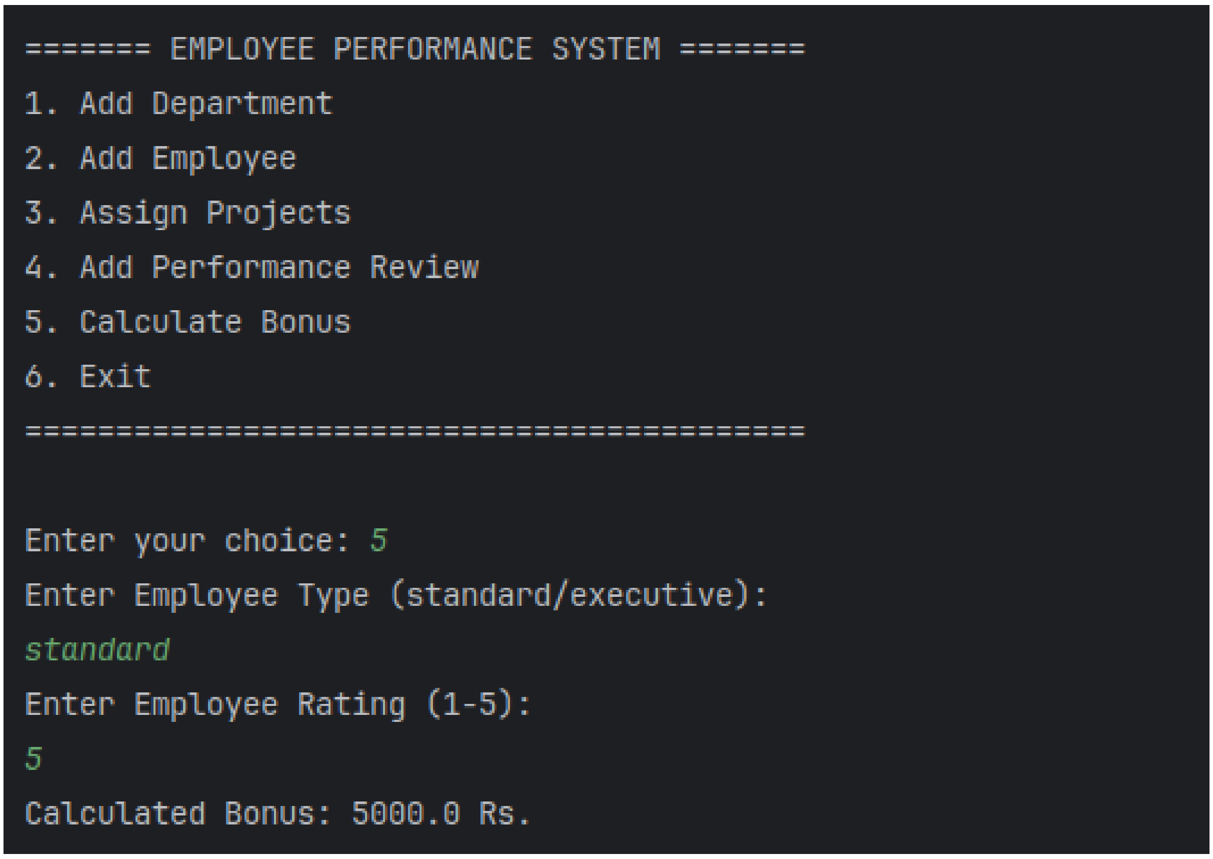
```
======= EMPLOYEE PERFORMANCE SYSTEM =======
1. Add Department
2. Add Employee
3. Assign Projects
4. Add Performance Review
5. Calculate Bonus
6. Exit
===========================================

Enter your choice: 4
Enter Employee ID for Performance Review:
1
Enter Performance Rating (1-5):
4
Enter Review Date (YYYY-MM-DD):
2025-06-20
Enter Review Remarks:
Excellent
Hibernate: select e1_0.id,d1_0.id,d1_0.name,e1_0.name,e1_0.rating,e1_0.salary from employee e1_0 left join department d1_0 on d1_0.id=e1_0.department_id
Hibernate: insert into performance_review (employee_id,rating,remarks,review_date) values (?,?,?,?)
Performance Review added successfully...
```



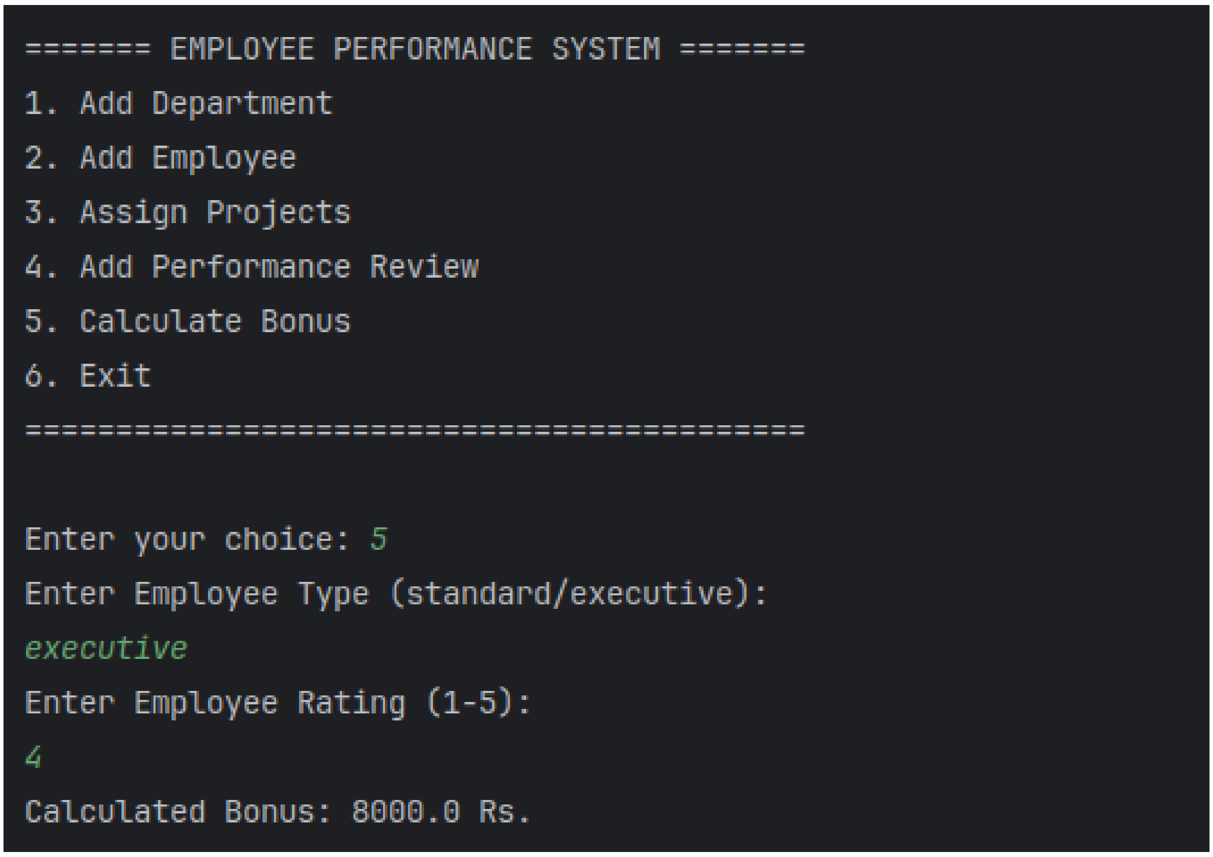
```
select * from performance_review;
```

	employee_id integer	id [PK] integer	rating integer	review_date timestamp without time zone (6)	remarks character varying (255)
1	1	1	4	2025-06-20 00:00:00	Excellent

5. Calculate Bonus

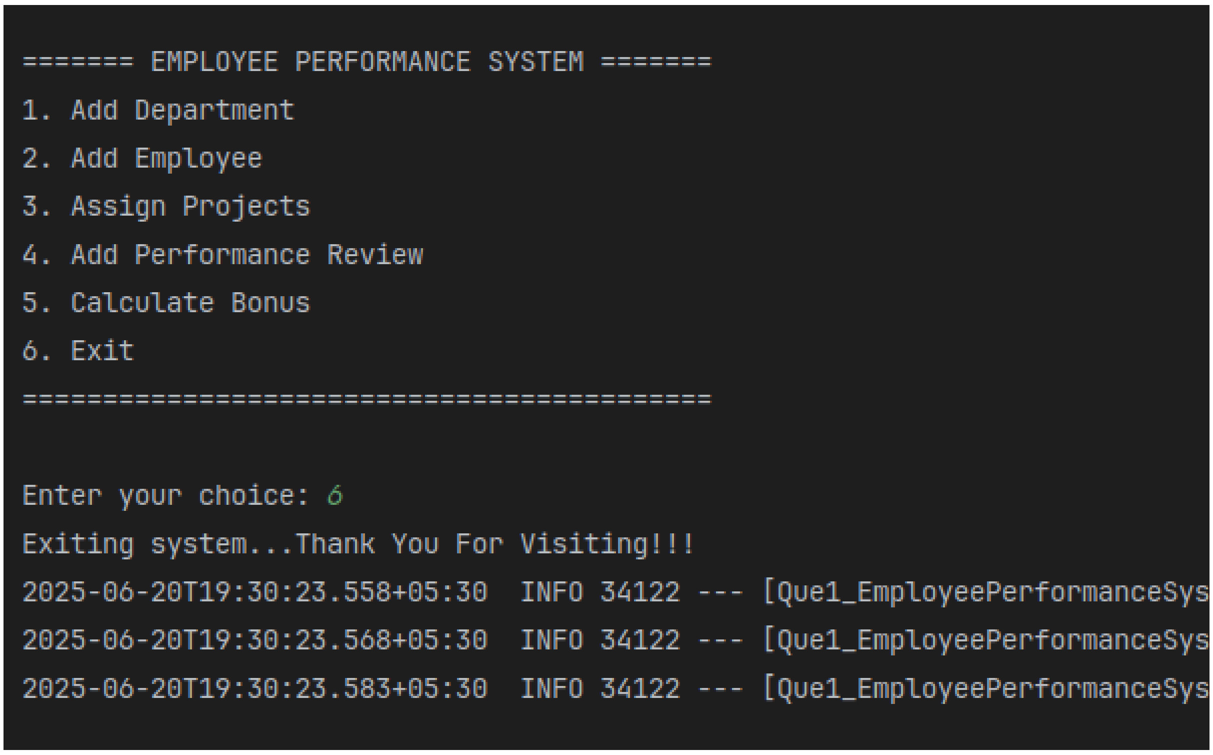


```
======= EMPLOYEE PERFORMANCE SYSTEM =======
1. Add Department
2. Add Employee
3. Assign Projects
4. Add Performance Review
5. Calculate Bonus
6. Exit
===========================================


Enter your choice: 5
Enter Employee Type (standard/executive):
standard
Enter Employee Rating (1-5):
5
Calculated Bonus: 5000.0 Rs.
```



```
======= EMPLOYEE PERFORMANCE SYSTEM =======
1. Add Department
2. Add Employee
3. Assign Projects
4. Add Performance Review
5. Calculate Bonus
6. Exit
===========================================


Enter your choice: 5
Enter Employee Type (standard/executive):
executive
Enter Employee Rating (1-5):
4
Calculated Bonus: 8000.0 Rs.
```

6. Exit



```
======= EMPLOYEE PERFORMANCE SYSTEM =======
1. Add Department
2. Add Employee
3. Assign Projects
4. Add Performance Review
5. Calculate Bonus
6. Exit
===========================================

Enter your choice: 6
Exiting system...Thank You For Visiting!!!
2025-06-20T19:30:23.558+05:30  INFO 34122 --- [Que1_EmployeePerformanceSys
2025-06-20T19:30:23.568+05:30  INFO 34122 --- [Que1_EmployeePerformanceSys
2025-06-20T19:30:23.583+05:30  INFO 34122 --- [Que1_EmployeePerformanceSys
```