// Employee.java

@Entity

@Table(name = "employee")

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name = "em\_id")

private int id;

@Column(name = "em\_name")

private String name;

@Column(name = "em\_salary")

private double salary;

@ManyToOne

@JoinColumn(name = "em\_dp\_id")

private Department department;

// Getters, Setters, toString

}

// Department.java

@Entity

@Table(name = "department")

public class Department {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name = "dp\_id")

private int id;

@Column(name = "dp\_name")

private String name;

// Getters, Setters, toString

}

// EmployeeRepository.java

@Repository

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

@Query("SELECT AVG(e.salary) FROM Employee e")

double getAverageSalary();

@Query("SELECT AVG(e.salary) FROM Employee e WHERE e.department.id = :id")

double getAverageSalaryByDept(@Param("id") int id);

}

// EmployeeService.java

@Service

public class EmployeeService {

private static final Logger LOGGER = LoggerFactory.getLogger(EmployeeService.class);

@Autowired

private EmployeeRepository employeeRepository;

@Transactional(readOnly = true)

public double getAverageSalary() {

return employeeRepository.getAverageSalary();

}

@Transactional(readOnly = true)

public double getAverageSalaryByDepartment(int deptId) {

return employeeRepository.getAverageSalaryByDept(deptId);

}

}

// OrmLearnApplication.java

@SpringBootApplication

public class OrmLearnApplication implements CommandLineRunner {

private static final Logger LOGGER = LoggerFactory.getLogger(OrmLearnApplication.class);

@Autowired

private EmployeeService employeeService;

public static void main(String[] args) {

SpringApplication.run(OrmLearnApplication.class, args);

}

@Override

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

testGetAverageSalary();

testGetAverageSalaryByDept();

}

private void testGetAverageSalary() {

LOGGER.info("Start testGetAverageSalary");

double avg = employeeService.getAverageSalary();

LOGGER.debug("Average Salary of All Employees: {}", avg);

LOGGER.info("End testGetAverageSalary");

}

private void testGetAverageSalaryByDept() {

LOGGER.info("Start testGetAverageSalaryByDept");

int deptId = 1; // use valid id

double avg = employeeService.getAverageSalaryByDepartment(deptId);

LOGGER.debug("Average Salary in Department {}: {}", deptId, avg);

LOGGER.info("End testGetAverageSalaryByDept");

}

}