Demonstrate implementation of O/R Mapping

@ManyToOne, @JoinColumn, @OneToMany, FetchType.EAGER, FetchType.LAZY, @ManyToMany, @JoinTable, mappedBy

Relationships reference - https://www.baeldung.com/spring-data-rest-relationships

* **@ManyToOne →**

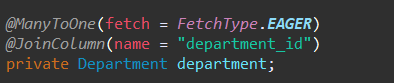
In Employee.java:

**Explanation**: Many employees belong to one department.

* **@JoinColumn**

in Employee.java, used to specify the foreign key column:

**Effect:** Creates a department\_id column in the employee table.

****

* **@OneToMany →**

In Department.java:

**Effect:** One department contains a list of employees.

* **FetchType.LAZY →**

Used on @OneToMany in Department.java:

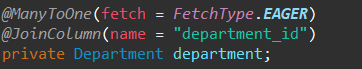
**Effect:** Employees of a department are loaded only when explicitly accessed.

****

* **FetchType.EAGER →**

Used on @ManyToOne in Employee.java:

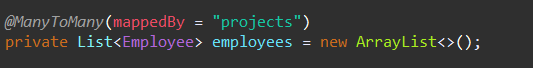
**Effect:** Department data is loaded automatically when Employee is fetched.

****

* **@ManyToMany**

In Employee.java and Project.java:

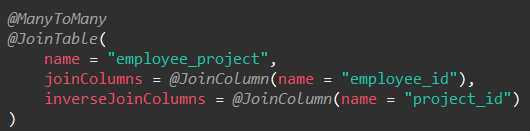
**Effect:** Employees and Projects have a many-to-many relationship.



* **@JoinTable →**

In Employee.java

**Effect:** Defines the join table for many-to-many relation.



* **mappedBy →**

In Project.java (inverse side):

**Effect:** Avoids creating a second join table and sets Employee as the owner of the relationship.

