OneToMany

@Entity  
public class Employee {  
 @OneToMany  
 List<Laptop> laptops;  
   
 @Id  
 @GeneratedValue  
 private Long id;  
  
 String name;  
  
 Double marks;  
  
}

One student have many employees, since in student tables as we cant have a list type,

Internally hibernate will create a separate table,

|  |  |
| --- | --- |
| Student\_id | Laptop\_id |
| S1 | L1 |
| S1 | L2 |

The above states, each student will have many laptops (s1 having 2 laps called l1,l2)

To stop this additional tables, we have to tell hibernate saying, BRO don’t create another table

Instead in laptop column itself we will create student id, use that , then we have to tell mapped by field

@OneToMany(mappedBy = "employee")  
List<Laptop> laptops;

## Mapping statements

1. One category will have many products & Many products belongs same category (here think in database row level)
2. One dept will have many student, each student belongs to one dept

## Cascade types

CascadeType.aLL- means whern we save Product , then save dependent Category object also to db

public class Product {  
 @ManyToOne(cascade = CascadeType.*ALL*)  
 @JoinColumn(name = "category\_id")  
 private Category category;

@JoinColumn

**in employee class , due to @OneToOne, it will create a foreign key column named (default is dependent class entityname+dep class pk col name- address\_add\_id),**

**if u dont want default name and if u want only custome fk col name, then give that name in @joinColumn("fk\_add\_id")**

**now that new fk in emp table as fk\_add\_id**

