

To retrieve data from database, we use either get method or load method

Usually get method is eager fetch and load is lazy fetch

Get method used to retrieve data with eager fetch	<pre> Session session1=sf.openSession(); Transaction tr1=session1.beginTransaction(); System.out.println("before get method"); pp1=session1.get(Product.class,1); System.out.println(pp1.getPname()); </pre>	Before get method Query will be executed Product name will be printed
Load is always used to retrieve data with lazy fetch	<pre> Session session1=sf.openSession(); Transaction tr1=session1.beginTransaction(); System.out.println("before get method"); Product pp1=session1.load(Product.class,2); System.out.println("After get method"); System.out.println(pp1.getPname()); </pre>	before get method After get method Query will be executed ,because of getter method at this point load will execute actual query Display name

@OneToOne

OneToOne annotation is by default eager fetch

And hence no matter if you use get /load in both cases join query will get executed,

But if we want to avoid join query to retrieve course, then we need to add lazy fetch type in OneToOne annotation.

<pre> private String name; private String address; @OneToOne(fetch=FetchType.LAZY) //eager fetch @JoinColumn(name="courseid") private Course c1; public Faculty() { super(); } </pre>	If the fetch type is set to lazy as shown, then always join will be avoided, and only faculty object will be retrieved, but if we use getter method of course f.getC1() , then it will select the course from the table

Default FetchType for relations

OneToOne ---eager

OneToMany ---Lazy

ManyToOne----eager

ManyToMany----lazy