**@ManyToOne🡪Unidirectional**

In the Student Entity🡪name, id, livingCountry(foreign key on Country table countryId column)

In the Country Entity🡪 countryId, name

Now Student to Country is many to one i.e **MANY** student lives in **A** country.

From Country to Student is one to many i.e in **A** country **MANY** student lives.

So under Student Entity

@ManyToOne(cascade=CascadeType.ALL)

Country studentCountry; // No Join Column needed.

Create Country, associate it with Student. Save Student. Saving the student will save Country also. It is because of Cascade

**@ManyToOne🡪Bidirectional**

In the Country Entity

@OneToMany(cascade=CascadeType.ALL, mappedBy=”studentCountry”).

Set<Student> studentList = new HashSet<Student>();

Note: mappedBy is having the same attribute name declared in @ManyToOne

The reason why mappedby is applicable only on Country Entity is 🡪 Country is Parent and Student is child🡪 child has physical foreign key which refers parent table.

🡪Parent doesn’t know about children. So we are saying in parent entity, parent is mapped to child by “mappedby”, whereas

**If mappedby is not given, hibernate will create another table for reference.**

<https://www.youtube.com/watch?v=ScPJ0SBkAsE>